Novel Restoration Techniques for Images Corrupted with High Density Impulsive Noise

A thesis submitted in fulfillment of the

requirements for the degree of Doctor of Philosophy

by

Ramesh Kulkarni

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela, INDIA

2012

Novel Restoration Techniques for Images Corrupted with High Density Impulsive Noise

A thesis submitted in fulfillment of the

requirements for the degree of Doctor of Philosophy

by

Ramesh Kulkarni

(Roll No. 50609002)

Under the supervision of

Prof Sukadev Meher

Prof J M Nair

Department of Electronics and Communication Engineering

National Institute of Technology, Rourkela, INDIA

2012

CERTFICATE

*This is to certify that the thesis titled "**Novel Restoration Techniques for Images Corrupted with High Density Impulsive Noise**", submitted to the National Institute of Technology, Rourkela (INDIA) by **Ramesh Kulkarni**, Roll No. **50609002** for the award of the degree of **Doctor of Philosophy in Electronics and Communication Engineering**, is a bona fide record of research work carried out by him under our supervision and guidance.*

The candidate has fulfilled all the requirements.

The thesis, which is based on candidate's own work, has not been submitted elsewhere for a degree/diploma.

In our opinion, the thesis is of standard required of a PhD degree in Engineering.

To the best of our knowledge, Mr. Ramesh Kulkarni bears a good moral character and decent behavior.

Prof Sukadev Meher
Professor & HOD, EC
NIT Rourkela

Prof J. M. Nair
Principal
VESIT, Mumbai

PREFACE

Digital Image Processing, developed during last three decades, has become a very important subject in all fields of engineering. Image filtering is one of the prime areas of image processing and its objective is to recover an image when it is corrupted with noise.

Impulsive noise is frequently encountered during the processes of acquisition, transmission and reception, and storage and retrieval. Usually median or a modified version of median is employed to suppress an impulsive noise. It is clear from the literature that the detection followed by filtering achieves better performance than the filters without detection. The noisy pixels are then replaced with estimated values. In this thesis, efforts are made to develop efficient filters for suppression of impulse noise under medium and high noise density conditions.

Two models of impulsive noise are considered in this thesis. The first one is *Salt-and-Pepper Noise* (SPN) model, where the noise value may be either the minimum or maximum of the dynamic gray-scale range of the image. And, the second one is *Random-Valued Impulsive Noise* (RVIN) model, where the noise pixel value is bounded by the range of the dynamic gray-scale of the image. Some proposed schemes deal with SPN model of noise as well as RVIN, whereas some other proposed schemes deal with only SPN. A few schemes are also proposed for color image denoising. The filters are tested on low, medium and high noise densities and they are compared with some existing filters in terms of objective and subjective evaluation. There are a number of filters available at low and medium noise densities, but they fail to perform at high noise densities. Therefore, there is sufficient scope to explore and develop efficient filters for suppressing the impulsive noise at high noise densities. Hence efforts are made here to develop efficient filters for suppression of impulse noise for medium and high noise densities. The execution time is taken into account while developing the filters for online and real-time applications such as digital camera, television, photo-phone, etc.

I hope the proposed filters in this thesis are helpful for other researchers working in this field for developing much better filters.

Ramesh Kulkarni

ACKNOWLEDGEMENT

I express my indebtedness and gratefulness to my teacher and supervisor **Dr. Sukadev Meher,** Professor & Head, Department of Electronics & Communication Engineering, for his continuous encouragement and guidance. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth. I am obliged to him for his moral support through all the stages during this doctoral research work. I am indebted to him for the valuable time he has spared for me during this work.

I am grateful to my co-supervisor **Prof. Jayalekshmi Nair**, Principal, VESIT, Mumbai, for her timely comments, guidance and support throughout the course of this work.

I am very much indebted to **Prof. S. K. Patra,** Chairman of DSC, who provided all the official facilities and guidance to me. I am also grateful to other DSC members, **Prof. Samit Ari** and **Prof. Dipti Patra** for their continuous support during the doctoral research work.

I would like to thank all my colleagues and friends, Prof. Shobha Krishnan, C.S.Rawat, N.Bhoi, M.Gupta, S.K.Dandpat and Ajit Sahoo for their company and cooperation during this period.

I take this opportunity to express my regards and obligation to my father and other family members whose support and encouragement I can never forget in my life.

I would like to thank my wife Anu and daughters Shrilaxmi and Shreya for their patience and cooperation. I can't forget their help who have managed themselves during the tenure of my Ph.D. work. I duly acknowledge the constant moral support they provided throughout.

Lastly, I am thankful to all those who have supported me directly or indirectly during the doctoral research work.

Ramesh Kulkarni

BIO-DATA OF THE CANDIDATE

Name of the candidate	:	Ramesh Kulkarni
Father's Name	:	Kushalrao Kulkarni
Date of Birth	:	11-11-1961
Present Address	:	(i) PhD Scholar, Dept. of Electronics and Communication Engg. National Institute of Technology Rourkela Rourkela-769 008 (India) (ii) Associate Professor Dept. of Electronics and Communication, V.E.S. Institute of Technology, Mumbai-400 074 (India)
Permanent Address	:	Plot no. A-33, Sector-7, Khanda Colony, New-Panvel (W) Panvel-410 206 (India)

ACADEMIC QUALIFICATION :

(i) **B. E.** in Electronic & Tele-Communication , BIET, Davangere, *Mysore University*, INDIA
(ii) **M. E.** in Digital Electronic , BVBCET, Hubli, *Karnataka University* INDIA

PUBLICATION:

(i) Published 04 papers in International Journals;
(ii) Communicated 02 papers to International Journals;
(iii) Published 11 papers in National and International Conferences.

Contents

Abstract

Impulse noise is a most common noise which affects the image quality during acquisition or transmission, reception or storage and retrieval process. Impulse noise comes under two categories: (1) fixed-valued impulse noise, also known as salt-and-pepper noise (SPN) due to its appearance, where the noise value may be either the minimum or maximum value of the dynamic gray-scale range of image and (2) random-valued impulse noise (RVIN), where the noisy pixel value is bounded by the range of the dynamic gray-scale of the image.

In literature, many efficient filters are proposed to suppress the impulse noise. But their performance is not good under moderate and high noise conditions. Hence, there is sufficient scope to explore and develop efficient filters for suppressing the impulse noise at high noise densities. In the present research work, efforts are made to propose efficient filters that suppress the impulse noise and preserve the edges and fine details of an image in wide range of noise densities.

It is clear from the literature that detection followed by filtering achieves better performance than filtering without detection. Hence, the proposed filters in this thesis are based on detection followed by filtering techniques.

The filters which are proposed to suppress the SPN in this thesis are:

- *Adaptive Noise Detection and Suppression* (***ANDS*****)** *Filter*
- *Robust Estimator based Impulse-Noise Reduction* (***REIR***) *Algorithm*
- *Impulse Denoising Using Improved Progressive Switching Median Filter* (***IDPSM***)
- *Impulse-Noise Removal by Impulse Classification* (***IRIC***)
- *A Novel Adaptive Switching Filter-I* (***ASF-I***) *for Suppression of High Density SPN*
- *A Novel Adaptive Switching Filter-II* (***ASF-II***) *for Suppression of High Density SPN*
- *Impulse Denoising Using Iterative Adaptive Switching Filter* (***IASF***)

In the first method, ANDS, neighborhood difference is employed for pixel classification. Controlled by binary image, the noise is filtered by estimating the value of a pixel with an adaptive switching based median filter applied exclusively to neighborhood pixels that are labeled noise-free. The proposed filter performs better in retaining edges and fine details of an image at low-to-medium densities of fixed-valued impulse noise.

The REIR method is based on robust statistic technique, where adaptive window is used for pixel classification. The noisy pixel is replaced with Lorentzian estimator or average of the previously processed pixels. Because of adaptive windowing technique, the filter is able to suppress the noise at a density as high as 90%.

In the proposed method, IDPSM, the noisy pixel is replaced with median of uncorrupted pixels in an adaptive filtering window. The iterative nature of the filter makes it more efficient in noise detection and adaptive filtering window technique makes it robust enough to preserve edges and fine details of an image in wide range of noise densities.

The forth proposed method is IRIC. The noisy pixel is replaced with median of processed pixels in the filtering window. At high noise densities, the median filtering may not be able to reject outliers always. Under such circumstances, the processed left neighboring pixel is considered as the estimated output. The computational complexity of this method is equivalent to that of a median filter having a 3×3 window. The proposed algorithm requires simple physical realization structures. Therefore, this algorithm may be quite useful for online and real-time applications.

Two different adaptive switching filters: ASF-I and ASF-II are developed for suppressing SPN at high noise density. The noisy pixel is replaced with *alpha-trimmed mean* value of uncorrupted pixels in the adaptive filtering window. Depending on noise estimation, a small filtering window size is initially selected and then the scheme adaptively changes the window size based on the number of *noise-free* pixels. Therefore, the proposed method removes the noise much more effectively even at noise density as high as 90% and yields high image quality.

In the proposed method IASF, noisy pixel is replaced with *alpha-trimmed mean* value of uncorrupted pixels in the adaptive filtering window. Due to its iterative structure, the performance of this filter is better than existing order-statistic filters. Further, the adaptive filtering window makes it robust enough to preserve the edges and fine details of an image.

The filters which are proposed for suppressing random-valued impulse noise (RVIN) are:

- *Adaptive Window based Pixel-Wise MAD* (***AW-PWMAD***) *Algorithm*
- *Adaptive Local Thresholding with MAD* (***ALT-MAD***) *Algorithm*

The proposed method, Adaptive Window based Pixel-Wise MAD (AW-PWMAD) Algorithm is a modified MAD (Median of the Absolute Deviations from the median) scheme alongwith a threshold employed for pixel-classification. The noisy pixel is replaced with median of uncorrupted pixels in adaptive filtering window.

Another proposed method for denoising the random-valued and fixed-valued impulse noise is ALT-MAD. A modified MAD based algorithm alongwith a local adaptive threshold is utilized for pixel-classification. The noisy pixel is replaced with median of uncorrupted pixels in the filtering window of adaptively varied size.

Three threshold functions are suggested and employed in this algorithm. Thus, three different versions, namely, **ALT-MAD-1, ALT-MAD-2** and **ALT-MAD-3** are developed. They are observed to be quite efficient in noise detection and filtering.

In the last part of the thesis, some efforts are made to develop filters for color image denoising. The filters which perform better in denoising gray-scale images are developed for suppression of impulsive noise from color images. Since the performance of denoising filters degrades in other color spaces, efforts are made to develop color image denoising filters in RGB color space only in this research work.

The developed filters are:

- *Multi-Channel Robust Estimator based Impulse-Noise Reduction* (**MC-*REIR*)** *Algorithm*
- *Multi-Channel Impulse-Noise Removal by Impulse Classification* **(MC-IRIC)**
- *Multi-Channel Iterative Adaptive Switching Filter* **(MC-IASF)**
- *Multi-Channel Adaptive Local Thresholding with MAD* **(MC-ALT-MAD)** *Algorithm*

It is observed from the simulation results that the proposed filters perform better than the existing methods. The proposed methods: ASF-1 and IASF exhibit quite superior performance in suppressing SPN in high noise densities compared to other methods. Similarly ALT-MAD-3 exhibits much better performance in suppressing RVIN of low to medium noise densities.

List of Abbreviations used

General Terminology

1.	*AWGN*	Additive White Gaussian Noise
2.	*SPN*	Salt-and-Pepper Noise
3.	*RVIN*	Random–Valued Impulse Noise
4.	*SN*	Speckle Noise
5.	*med*	Median
6.	*min, max*	Minimum, Maximum
7.	*MSE*	Mean Squared Error
8.	*MAE*	Mean Absolute Error
9.	*RMSE*	Root Mean Squared Error
10.	*MMSE*	Minimum Mean Squared Error
11.	*PSNR*	Peak Signal to Noise Ratio
12.	*CPSNR*	Color Peak Signal to Noise Ratio
13.	*UQI*	Universal Quality Index
14.	*IEF*	Image Enhancement Factor
15.	*MAD*	Median of the Absolute Deviations from the median
16.	*PWMAD*	Pixel-Wise MAD
17.	*HVS*	Human Visual System
18.	*CF*	Classifier Filter
19.	*SF*	Switching Filter
20.	*BCS*	Basic Classifier Filter
21.	*ICF-1*	Iterative Classifier-Filter-1
22.	*ICF-2*	Iterative Classifier-Filter-2

Filters (available in literature)

23.	*MF*	Mean Filter
24.	*ATM*	Alpha Trimmed Mean
25.	CWM	Center Weighted Median Filter
26.	TSM	Tri-State Median
27.	*AMF*	Adaptive Median Filter
28.	*PSM*	Progressive Switching Median Filter
29.	*SMF*	Switching Median Filter
30.	*AMAD*	Adaptive MAD
31.	*BDND*	Boundary Discrimination Noise Detection

Proposed Filters

32.	*ANDS*	Adaptive Noise Detection and Suppression Filter
33.	*REIR*	Robust Estimator based Impulse-Noise Reduction Algorithm
34.	*IDPSM*	Impulse Denoising Using Improved Progressive Switching Median Filter
35.	*IRIC*	Impulse Noise Removal in Highly Corrupted Image by Impulse Classification
36.	*ASF-I*	Adaptive Switching Filter-I
37.	*ASF-II*	Adaptive Switching Filter-II
38.	*IASF*	Impulse Denoising Using Iterative Adaptive Switching Filter
39.	*AW-PWMAD*	Adaptive Window based Pixel-Wise MAD Algorithm
40.	*ALT-MAD*	Adaptive Local Thresholding with MAD Algorithm
41.	*MC-REIR*	Multi-Channel Robust Estimator based Impulse-Noise Reduction Algorithm
42.	MC-*IRIC*	Multi-Channel Impulse-Noise Removal by Impulse Classification
43.	*MC-IASF*	Multi-Channel Iterative Adaptive Switching Filter
44.	*MC-ALT-MAD*	Multi-Channel Adaptive Local Thresholding with MAD Algorithm

List of Symbols used

Symbols

1.	$f(i,j)$	Original (noise-free) digital image with discrete spatial coordinates (i, j)
2.	f_{min}	Minimum value of pixels
3.	f_{max}	Maximum value of pixels
4.	$g(i,j)$	Noisy (input) image
5.	g_{min}	Minimum pixel value in a window
6.	g_{max}	Maximum pixel value in a window
7.	η	Random Variable; Noise
8.	T_E	Execution Time
9.	$\hat{f}(i,j)$	Filtered (output) image
10.	$b(i,j)$	Binary image
11.	$g_{k,l}(i,j)$	Windowed (sampled) input image, i.e., a sub-image
12.	M, N	Number of rows (columns) of an image matrix
13.	P, Q	Number of rows (columns) of a sub-image
14.	$M_{k,l}(i,j)$	Mapped image in a window
15.	$D_{k,l}(i,j)$	Difference image in a window
16.	$m(i,j)$	Median of a window
17.	m	Median of whole image
18.	$d(i,j)$	Absolute deviation image
19.	K	Kernel (for Laplacian operator)
20.	K_p	p^{th} kernel
21.	C_1	Count of noisy pixels in an image
22.	C_2	Count of noise-free pixels in an image
23.	γ	Noise density observed, i.e., $C_1/(C_1+C_2)$
24.	T	Threshold (fixed)
25.	β	Threshold (adaptive)
26.	$\psi(x_l)$	Influence function
27.	ρ (.)	Lorentzian estimator
28.	σ	Outlier rejection point
29.	τ_s	Maximum expected outlier
30.	σ_N	Standard deviation
31.	$\boldsymbol{\zeta}$	Smoothening factor
32.	Cw_2	Count of noise-free pixels in selected window
33.	$\mathbb{Z}^+$	Set of integers
34.	$\mathbb{U}$	MAD
35.	$\mathbb{V}$	PWMAD
36.	$d_{\boldsymbol{k}}$	Absolute Deviation from Median
37.	k	k = (i,j), a vector index representing elements in a selected window

Chapter 1

Introduction

1

Preview

The aim of digital image processing is to improve the potential information for human interpretation and processing of image data for storage, transmission, and representation for autonomous machine perception. The quality of image degrades due to contamination of various types of noise. Additive white Gaussian noise, Rayleigh noise, Impulse noise etc. corrupt an image during the processes of acquisition, transmission and reception and storage and retrieval. For a meaningful and useful processing such as image segmentation and object recognition, and to have very good visual display in applications like television, photo-phone, etc., the acquired image signal must be noise-free and made deblurred. Image deblurring and image denoising are the two sub-areas of image restoration. In the present research work, efforts are made to propose efficient filters that suppress the noise and preserve the edges and fine details of an image as far as possible in wide range of noise density.

The following topics are covered in this chapter.

- *Fundamentals of Digital Image Processing*
- *Noises in Digital Images*
- *Literature Review*
- *Problem Statement*
- *Basics of Spatial Filtering*
- *Image Metrics*
- *Chapter-wise Organization of the Thesis*
- *Conclusion*

1.1 Fundamentals of Digital Image Processing

A major portion of information received by a human being from the environment is visual. Hence, processing visual information by computer has been drawing a very significant attention of the researchers over the last few decades. The process of receiving and analyzing visual information by the human species is referred to as sight, perception and understanding. Similarly, the process of receiving and analyzing visual information by digital computer is called *digital image processing* [1].

An image may be described as a two-dimensional function $f(i,j)$, where *i* and *j* are spatial coordinates. Amplitude of *f* at any pair of coordinates (x,y), is called intensity or gray value of the image. When spatial coordinates and amplitude values are all finite, discrete quantities, the image is called *digital image* [2]. Each element of this matrix (2-D array) is referred as *picture element* or *pixel*. Image Processing (IP) is a branch of study where a 2-D image signal $f(i,j)$ is processed either directly (spatial-domain processing) or indirectly (transform-domain processing). IP and Computer vision are two separate fields with a narrow boundary between them. In case of IP, both input and output are 2-D images whereas the output of a Computer vision system is necessarily not an image rather some attributes of it.

In computer vision, the ultimate goal is to use computer to emulate human vision, including performing some analysis, judgment or decision making or performing some mechanical operation (robot motion) [11-14]. Fig. 1.1 shows a typical image processing system [1, 2].

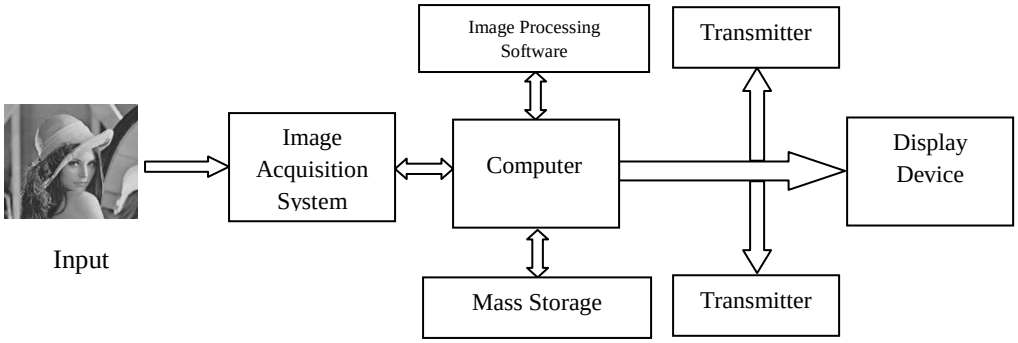


Fig. 1.1 Basic Block Diagram

Following is the list of most common image processing functions.

- *Image Representation*
- *Image Transformation*
- *Image Enhancement*
- *Image Restoration*
- *Color Image Processing*
- *Transform-Domain Processing*
- *Image Compression*
- *Morphological Image Processing*
- *Image Representation and Description*
- *Object Recognition*

For the first seven functions, the inputs and outputs are images whereas for the rest three the outputs are attributes of the input images. With the exception of image acquisition and display, most image processing functions are usually implemented in software. Image processing is characterized by specific solutions; hence the technique that works well in one area may be inadequate in another.

Image processing begins with an image acquisition process. The two elements are required to acquire digital images. The first one is a sensor; it is a physical device that is sensitive to the energy radiated by the object that has to be imaged. The second part is called a digitizer. It is a device for converting the output of the sensing device into digital form. For example in a digital camera, the sensors produce an electrical output proportional to light intensity. The digitizer converts the outputs to digital data. During the process of image acquisition noises are introduced.

Image processing may be performed in spatial or transform-domain. Different transforms (e.g. Discrete Fourier Transform (DFT) [1], Discrete Cosine Transform (DCT) [14, 16], Discrete Hartley Transform (DHT) [21], Discrete Wavelet Transform (DWT) [9-13, 17-20, 22], etc., are used for different applications.

Image enhancement is among the simplest and most appealing areas of digital image processing [114-120]. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. A familiar example of enhancement is when we increase the contrast of an image *it looks better*. It is important to keep in mind that image enhancement is a subjective area of image processing. On the other hand, image restoration is very much objective. The restoration techniques are based on mathematical and statistical models of image degradation. *Denoising* [121-133] and *deblurring* tasks come under this category.

Image restoration and filtering is one of the prime areas of image processing and its objective is to recover the images from degraded observations. The techniques involved in image restoration and filtering are oriented towards modeling the degradations and then applying an inverse operation to obtain an approximation of the original image. The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from scene. Second, humans can discern thousands of color shades and intensities, compared to shades of gray.

The first encounter with digital image restoration in the engineering community was in the area of astronomical imaging during 1950s and 1960s. The aim of the mission was to record many incredible images of solar system. However, the images obtained from the various planetary missions of the time were subject to much photographic degradation. This mission required huge amount of money. The degradations occurred due to substandard imaging environment, rapidly changing refractive index of the atmosphere and slow camera shutter speed relative to spacecraft. Any loss of information due to image degradation was devastating as it reduced the scientific value of these images.

In the area of medical imaging, image restoration has certainly played a very important role. Restoration has been used for filtering noise in X-ray, mammograms, and digital angiographic images.

Another application of this field is the use of digital techniques to restore ageing and deteriorated films. The idea of motion picture restoration is probably most often associated with the digital techniques used not only to eliminate scratches and dust from celluloid films of old movies, but also to colorize black-and-white (gray-scale) films.

Digital image restoration is being used in many other applications as well. Just to name a few, restoration has been used to restore blurry X-ray images of aircraft wings to improve quality assessment procedures. It is used for restoring the motion induced effects present in still composite frames and more generally, for restoring uniformly blurred television pictures. Digital restoration is also used to restore images in automated assembly / manufacturing process. Many defense-oriented applications require restoration, such as guided missiles, which may obtain distorted images due to the effects of pressure differences around a camera mounted on the missile.

Digital images, which are 2-D signals, are often corrupted with many types of noise, such as *additive white Gaussian noise* (AWGN) which is referred as *additive noise* and *substitutive noise* such as, *salt-and-pepper noise* (SPN), *random-valued impulse noise* (RVIN), *multi-level noise* during the processes of acquisition, transmission and reception, and storage and retrieval. The impulse noise is substitutive noise, i.e. the corrupted pixel value does not depend on the original pixel value, whereas additive Gaussian noise modifies the original pixel value with uniform power in the whole bandwidth and with Gaussian probability distribution. Impulse noise comes under two categories: (1) fixed-valued impulse noise and (2) random-valued impulse noise. Under fixed-valued impulse noise, the noise may be unipolar or bipolar. In many occasions an image is observed to be corrupted with bipolar fixed value impulse noise. A fixed-valued bipolar impulse noise is called salt-and-pepper noise (SPN) due to its appearance. The malfunctioning pixels in camera sensors, faulty memory location in hardware, or transmission of the image in a noisy channel, are the some of the common causes for impulse noise [38, 58-61]. The intensity of impulse noise has the tendency of either being relatively high or low. Due to this,

when the signal is quantized to 'L' intensity levels, the corrupted pixels are generally digitized into either minimum or maximum values in the dynamic range, these pixels appear as white or black dots in the image. This may severely degrade the image quality and cause some loss of image information. Keeping the image details and removing the noise from the digital image is a challenging part of image processing [29, 66-86].

It is difficult to suppress AWGN since it corrupts almost all pixels in an image. The arithmetic mean filter, commonly known as Mean filter [37-39], can be employed to suppress AWGN but it introduces a *blurring* effect [16-20, 22]. Efficient suppression of noise in an image is a very important issue. Conventional techniques of image denoising using linear and nonlinear techniques have already been reported and sufficient literatures are available in this area [1-6, 23-42].

A number of nonlinear and adaptive filters are proposed for denoising an image. The aim of these filters is to reduce the noise as well as to retain the edges and fine details of the images [23-28, 124-128]. But it is difficult to achieve both the objectives and the reported schemes are not able to perform in both aspects. Hence, still various research workers are actively engaged in developing better filtering schemes using latest signal processing techniques. The present doctoral research work is focused on developing quite efficient image denoising filters to suppress Impulse Noise quite effectively without yielding much distortion and blurring.

1.2 Noise in Digital Images

In this section, various types of noise corrupting an image signal are studied; the types of noise are discussed, and mathematical models for the different types of noise are presented.

1.2.1 Types of Noise

The principal sources of noise in digital images arise during image acquisition and/or transmission. The performance of image sensors is affected by a variety of factors such as environmental conditions during image acquisitions, and quality of sensing elements themselves. Images are corrupted during transmission principally due to electromagnetic interference in a channel employed for transmission. For example, an

image transmitted using a wireless network might be corrupted because of lightening or other atmospheric disturbances.

When an analog image signal is transmitted through a linear dispersive channel, the image edges (step-like or pulse like signal) get blurred and the image signal gets contaminated with AWGN since no practical channel is noise free. If the channel is so poor that the noise variance is high enough to make the signal excurse to very high positive or high negative value, then the thresholding operation at the front end of the receiver will contribute saturated max and min values. Such noisy pixels will be seen as white and black spots in the image. Therefore, this type of noise is known as salt-and-pepper noise (SPN). So, if analog image signal is transmitted, then the signal gets corrupted with AWGN and SPN as well. Thus, there is an effect of mixed noise [158].

If the image signal is transmitted in digital form through a linear dispersive channel, then inter-symbol interference (ISI) takes place. In addition to this, the AWGN in a practical channel also comes into picture. This makes the situation very critical. Due to ISI and AWGN, it may so happen that a '1' may be recognized as '0' and vice-versa. Under such circumstances, the image pixel values have changed to some random values at random positions in the image frame. Such type of noise is known as random-valued impulse noise (RVIN).

Another type of noise that may corrupt an image signal is the *speckle noise* (SN). In some biomedical applications like ultrasonic imaging and a few engineering applications like synthesis aperture radar (SAR) imaging, such a noise is encountered. The SN is a signal dependent noise, i.e., if the image pixel magnitude is high, then the noise is also high. The noise is multiplicative because initially a transmitting system transmits a signal to the object and the reflected signal is recorded. When the signal is transmitted, the signal may get contaminated with additive noise in the channel. Due to varying reflectance of the surface of the object, the reflected signal magnitude varies. So also the noise varies since the noise is also reflected by the surface of the object. Noise magnitude is, therefore, higher when the signal magnitude is higher. Thus, the speckle noise is multiplicative in nature.

The speckle noise is encountered only in a few applications like ultrasonic imaging and SAR, whereas all other types of noise like, AWGN, SPN, and RVIN occur in almost all the applications.

1.2.2 Mathematical Models of Noise

There are different types of noises which corrupt an image. The noise like Gaussian Noise, Rayleigh Noise, Gamma Noise, Speckle Noise and Impulse Noise are quite common. A few important noise models are presented in this section.

Additive White Gaussian Noise:

Let $g(i,j)$ be a noisy image formed due to addition of noise $\eta(i,j)$ to an original image $f(i,j)$, which is represented as

$$g(i,j) = f(i,j) + \eta(i,j) \tag{1.1}$$

where, noise $\eta(i,j)$ is represented by a Gaussian Probability Density Function (PDF).

The PDF of Gaussian random variable, *t*, is given by

$$\frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \tag{1.2}$$

where, t is gray level; μ is mean value of t; and σ is its standard deviation.

When the variance, σ^2 of the random noise $\eta(t)$ is very low, then $\eta(i,j)$ is zero or very close to zero at many pixel locations. Under such circumstances, the noisy image $g(i,j)$ is same or very close to the original image $f(i,j)$ at many pixel locations(i,j).

Impulse Noise:

The SPN and RVIN, which are generally categorized as impulse noise, are substitutive in nature. The impulse noise occurs at random locations (i,j).

Let a digital image$f(i,j)$, after being corrupted with SPN of density *d* be represented as $g(i,j)$. Then, the noisy image $g(i,j)$ is mathematically represented as:

$$g(i,j) = \begin{cases} f(i,j) & probability,\ p = 1-d \\ 0 & probability,\ p = d/2 \\ 1 & probability,\ p = d/2 \end{cases} \tag{1.3}$$

If it is corrupted with RVIN of density *d*, it is mathematically represented as:

$$g(i,j) = \begin{cases} f(i,j) & probability,\ p = 1-d \\ \eta(i,j) & probability,\ p = d \end{cases} \quad (1.4)$$

Here, $\eta(i,j)$ represents a uniformly distributed random variable, ranging from 0 to 1, that replaces the original pixel value $f(i,j)$. The noise magnitude at any noisy pixel location (i,j) is independent of the original pixel magnitude. Therefore, the RVIN is truly substitutive.

Speckle Noise:

Let a digital image $f(i,j)$, after being corrupted with multiplicative noise, be represented as $g(i,j)$. Then, the noisy image $g(i,j)$ is mathematically represented as:

$$g(i,j) = f(i,j) + \eta(i,j)f(i,j) \quad (1.5)$$

$$g(i,j) = [1 + \eta(i,j)]f(i,j) \quad (1.6)$$

where, $\eta(t)$ is a random variable.

The proposed filters developed in subsequent chapters are meant for suppression of low to high density impulse noise.

1.3 Literature Review

Noise in an image is a serious problem. Efficient suppression of noise in an image is a very important issue. Denoising finds extensive applications in many fields of image processing. Conventional techniques of image denoising using linear and nonlinear filters have already been reported and sufficient literature is available in this area. Recently, various nonlinear and adaptive filters have been suggested for the purpose. The objectives of these schemes are to reduce noise and to retain, as far as possible, the edges and fine details of the original image in the restored image as well. However, both the objectives conflict each other and the reported schemes are not able to perform satisfactorily in both aspects. Hence, still various research workers are actively engaged in developing better filtering schemes using latest signal processing techniques.

1.3.1 Filters for Suppression of Additive Noise

Traditionally, AWGN is suppressed using linear spatial domain filters such as Mean filter [1-7], Wiener filter [1, 2, 8, 15, 40-42] etc. The traditional linear techniques are very simple in implementation but they suffer from disadvantage of blurring effect. They also don't perform well in the presence of signal dependant noise. To overcome

this limitation, nonlinear filters [4] are proposed. Some well known nonlinear mean filters are harmonic mean, geometric mean, L_p mean, contra-harmonic mean proposed by Pitas et al. [5] are found to be good in both preserving edges and suppressing the noise. Another good edge preserving filter is Lee filter [43] proposed by J.S. Lee. The performance of this filter is also good in suppressing noise as well as in preserve edges. Anisotropic diffusion [44, 45] is also a powerful filter where local image variation is measured at every point, and pixel values are averaged from neighborhoods whose size and shape depend on local variation. The basic principle of these methods is numbers of iterations. If more numbers of iterations are used it may lead to instability; in addition to edges, noise becomes prominent. Rudin *et al*. proposed total variation (TV) filter [46] which is also iterative in nature. In the later age of research, simple and non-iterative scheme of edge preserving smoothing filters are proposed. One of them is Bilateral filter [47]. Bilateral filter works on the principle of geometric closeness and photometric similarity of gray levels or colors. Many variants of Bilateral filters are proposed in literature that exhibit better performance under high noise condensation [48, 49]. A filter named non-local means (NL-Means) [50] averages similar image pixels defined according to their local intensity similarity. Based on robust statistics, a number of filters are proposed. T. Rabie [51] proposed a simple blind denoising filter based on the theory of robust statistics. Robust statistics addresses the problem of estimation when the idealized assumptions about a system are occasionally violated. Another denoising method based on the *bi-weight mid-regression* is proposed by Hou *et al*. [52] is found to be effective in suppressing AWGN. Kernel regression is a nonparametric class of regression method used for image denoising [53].

Many filters based on Fuzzy logic are developed for suppression of additive noise [36, 37, 54]. Ville *et al*. [54] proposed a fuzzy filter for suppression of AWGN. The first stage computes a fuzzy derivative for eight different directions. The second stage uses these fuzzy derivatives to perform fuzzy smoothing by weighting the contributions of neighboring pixel values. By applying iteratively the filter effectively reduces high noise.

Now-a-days, wavelet transform is employed as a powerful tool for image denoising [55-57]. Image denoising using wavelet techniques is effective because of

its ability to capture most of the energy of a signal in a few significant transform coefficients, when natural image is corrupted with Gaussian noise.

1.3.2 Filters for Suppression of Impulsive Noise

An impulsive noise of low and moderate noise densities can be removed easily by simple denoising schemes available in the literature. A simple median filter [58] works very nicely for suppressing impulsive noise of low density and is easy to implement. But the cost paid for it distorts edges and fine details of an image. The distortion increases as the filtering window size is increased to suppress high density noise. Specialized median filters such as weighted median filter [58-63, 86], center weighted median filter [64-66, 81, 82] and Recursive Weighted Median Filter (RWMF) [65] are proposed in literature to improve the performance of the median filter by giving more weight to some selected pixel(s) in the filtering window. But they are still implemented uniformly across an image without considering whether the current pixel is noisy or not. Additionally, they are prone to edge jitter in cases where the noise density is high. As a result, their effectiveness in noise suppression is often at the expense of blurred and distorted image features.

Conventional median filtering approach applies the median operation everywhere without considering whether it is uncorrupted or not. As a result, image quality degrades severely. An intuitive solution to overcome this problem is to implement an impulse-noise detection mechanism prior to filtering; hence, only those pixels identified as *corrupted* would undergo the filtering process, while those identified as *uncorrupted* would remain intact. By incorporating such noise detection mechanism or *intelligence* into the median filtering framework, so-called *switching median filters* [68, 69, 72-76, 79] have shown significant performance improvement. A number of modified median filters have been proposed [82-84], e.g., *minimum–maximum exclusive mean* (MMEM) filter [80] proposed by W.Y.Han *et al*., *pre-scanned minmax center-weighted* (PMCW) filter [81] proposed by Wang, and *decision-based median filter* [69] proposed by D.A.Florencio *et al*.. In these methods, the filtering operation adapts to the local properties and structures in the image. In the decision-based filtering [82-85] for example, image pixels are first classified as *corrupted* and *uncorrupted*, and then passed through the median and identity filters, respectively. The main issue of the decision-based filter lies in building a decision

rule, or a noise measure [106-109], that can discriminate the uncorrupted pixels from the corrupted ones as precisely as possible.

In MMEM filter [80]; where the pixels that have values close to the maximum and minimum in a filter window are discarded, and the average of remaining pixels in the window is computed to estimate a pixel. If the difference between the center pixel and average exceeds a threshold, the center pixel is replaced by average; otherwise, unchanged. The performance of this filter depends on the selection of threshold value. One simple switching filter Adaptive Center-Weighted Median (ACWM) [66] proposed by T.Chen *et al*, Center-Weighted Median (CWM) [64] has been used to detect noisy pixels in the first stage. The objective is to utilize the *center-weighted median* filters that have varied center weights to define a more general operator, which realizes the impulse detection by using the differences defined between the outputs of CWM filters and the current pixel of concern. The ultimate output is switched between the median and the current pixel itself. While still using a simple thresholding operation, the proposed filter yields superior results to other switching schemes in suppressing both types of impulses with different noise ratios. But its estimation efficiency is poor. Florencio *et al.* [69] proposed a decision measure, based on a second order statistic called normalized deviation.

The *peak and valley filter* [70] proposed by Windyga, is a highly efficient nonlinear non-iterative multidimensional filter. It identifies noisy pixels by inspecting their neighborhood, and then replaces their values with the most conservative ones out of the values of their neighbors. In this way, no new values are introduced into the neighborhood and the histogram distribution range is conserved. The main advantage of this filter is its simplicity and speed, which makes it very attractive for real time applications. A modified peak and valley filter, *detail preserving impulsive noise removal* [71] scheme has also been proposed by N. Alajlan. This filter provides better detail preservation performance; but it is slower than the original peak and valley filter.

The *tri-state median filter* [86] proposed by T.Chen *et al*, further improved switching median filters that are constructed by including an appropriate number of center-weighted median filters into the basic switching median filter structure. These filters exhibit better performance than the standard and the switching median filters at

the expense of increased computational complexity. Z.Wang *et al*. have proposed a *progressive switching median filter* (PSM) [72] for the removal of impulse noise from highly corrupted images where both the impulse detector and the noise filter are applied progressively in iterative manner. The noise pixels processed in the current iteration are used to help the process of the other pixels in the subsequent iterations. A main advantage of such a method is that some impulse pixels located in the middle of large noise blotches can also be properly detected and filtered. Therefore, better restoration results are expected, especially for the cases where the images are highly corrupted. A *new impulse noise detection technique* [73] for switching median filters proposed by S. Zhang *et al.* is based on the minimum absolute value of four convolutions obtained using one-dimensional Laplacian operators. It provides better performance than many of the existing switching median filters with comparable computational complexity.

Early developed switching median filters are commonly found being non adaptive to a given, but unknown, noise density and prone to yielding pixel misclassifications especially at higher noise density interference. To address this issue, the *noise adaptive soft-switching median* (NASM) filter is proposed H.L. Eng *et al.* [74], which consists of a three-level hierarchical soft-switching noise detection process. The NASM achieves a fairly robust performance in removing impulse noise, while preserving signal details across a wide range of noise densities, ranging from 10% to 50%. However, for those corrupted images with noise density greater than 50%, the quality of the recovered images become significantly degraded, due to the sharply increased number of misclassified pixels.

The *signal-dependent rank-ordered mean filter* [85] is a switching mean filter that exploits rank order information for impulse noise detection and removal. The structure of this filter is similar to that of the switching median filter except that the median filter is replaced with a *rank-ordered mean* of its surrounding pixels. This filter has been shown to exhibit better noise suppression and detail preservation performance than some conventional and state-of-the-art impulse noise cancellation filters for both grey scale [85] and color [34, 132-137] images.

The *adaptive two-pass rank order filter* [87] has been proposed by X.Xu, to remove impulse noise from highly corrupted images. Between the passes of filtering,

an adaptive process detects irregularities in the spatial distribution of the estimated noise and selectively replaces some pixels changed by the first pass with their original values. These pixels are kept unchanged during the second filtering. Consequently, the reconstructed image maintains a higher degree of fidelity and has a smaller amount of noise.

A variational approach to remove outliers and impulse noise [88] by M.Nikolova, is an edge and detail-preserving restoration technique to eliminate impulse noise efficiently. It uses a non-smooth data fitting term together with edge-preserving regularization functions. A combination of this variational method [88] with an impulse detector has also been presented in an iterative procedure for removing random-valued impulse noise [89]. The filter offers good filtering performance but its implementation complexity is higher than most of the previously mentioned filters.

The method proposed by I. Aizenberg *et al*. [90], employs boolean functions for impulse noise removal. In this approach, the gray level noisy input image is decomposed into a number of binary images by gray level thresholding. Detection and removal of impulse noise are then performed on these binary images by utilizing specially designed boolean functions. Finally, the resulting boolean images are combined back to obtain a restored grey level image.

A number of filters utilize the histogram information of the input image. In *image restoration using parametric adaptive fuzzy filter* [91] and *an adaptive fuzzy filter for restoring highly corrupted image by histogram estimation* [92], the histogram information of the input image is used to determine the parameters of the membership functions of an adaptive fuzzy filter. The filter is then used for the restoration of noisy images. An adaptive vector filter exploiting histogram information is also proposed for the restoration of color images [136].

With *boundary discriminative noise detection* (BDND) algorithm proposed by Pei-Eng Ng *et al*. [106], a highly-accurate noise detection algorithm, an image corrupted even up to 70% noise density may be restored quite efficiently. But there is no remarkable improvement in the results at higher noise density.

In addition to the median and the mean based filtering methods discussed above, a number of nonlinear impulse noise filtering operators based on soft

computing methodologies have also been presented [93-100]. These filters exhibit relatively better noise removal and detail preservation capability than the median and the mean based operators. However, the implementation complexities of these filters are generally too much and the required filtering window size is usually larger than the other methods. Indeed, neuro-fuzzy (NF) [101-105] systems inherit the ability of neural networks to learn from examples and derive the capability of fuzzy systems to model the uncertainty which is inevitably encountered in noisy environments. Therefore, neuro-fuzzy systems may be utilized to design line, edge, and detail preserving impulse noise removal operators provided that the appropriate network topologies and processing strategies are employed. The method proposed by Wenbin Luo *et al.* [113] uses a fuzzy classifier for pixel-classification and a simple median filter is employed for replacement of corrupted pixels. The methods proposed by F.Russo [30] and F. Farbiz *et al.* [31], uses neruo-fuzzy for filtering purpose.

In recent years, a number of methods have been proposed which work on both random-valued and salt-and-pepper noise [112,143-148]. The method proposed by V.Crnojevic *et al, Advanced Impulse Detection Based on Pixel-Wise MAD*, [122] is a modification of absolute deviation from median (MAD). MAD is used to estimate the presence of image details. An iterative pixel-wise modification of MAD is used here that provides a reliable removal of impulse noise. An improved method of this algorithm is *impulse noise filter with adaptive MAD based threshold* [129] proposed by Vladimir *et al*.. In this system the threshold value is changed from pixel to pixel based on local statistics. Since it is a non-iterative algorithm, its execution time is quite reasonable and less than that required by PWMAD. The performance of both the methods is quite good under low noise density. But they fail miserably at high noise densities. In the same category one more method proposed by Tzu–ChoLin is known as *progressive decision based mean type filter* [130]. This is based on Dempster-Shafer (D-S) evidence theory for pixel-classification. The mass functions are generated based on information available in the filtering window which are used for the D-S evidence theory. Decision rules can determine whether the pixel is noisy or not based on the noise-corrupted belief value. Both detection and filtering are applied progressively through several iterations. The corrupted pixels are replaced by the mean of the noise-free pixels in the filter window.

An efficient method developed by Jianjun Zhang [112] performs well for filtering random-valued noise. In this method, an adaptive center weighted median filter is used to identify pixels which are likely to be corrupted and restored by using median filter.

A simple iteration procedure is used for noise detection and filtering purpose. In Iterative *Adaptive Switching Median Filter* [110] proposed by S.Saudia *et. al,* a two-pass algorithm is employed for identification of a noisy pixel and replacing the corrupted pixel by a valid median. Another iterative filter is proposed by R.H.Chan *et al* [143] for effective suppression of random-valued noise. As it takes a large number of iterations, its execution time is too much. Further, it fails to retain the edges and fine details of an image at higher densities.

The method proposed by Haindi Ibrahim *et al*. [111] is an adaptive median filter to remove impulse noise from highly corrupted images. In fact, it is a hybrid of adaptive median filter with switching median filter. The adaptive median filter changes its size according to local noise density estimated. The switching framework helps to speedup the process of filtering. This method preserves the local details and edges of an image at medium noise densities. But there is no remarkable improvement in the results at higher noise densities.

Recently, a number of algorithms are proposed [138-142, 149-167, 172-174] for suppressing impulse noise. Different types of noise detection and correction techniques are proposed for filtering based on statistics, fuzzy logic and neural network. They work effectively; but, they fail to retain edges and fine details of an image at high noise densities even though they have high computational complexities. But, none of the filters available in literature is able to achieve very good restoration without distorting the edges and fine details. Further, there is a need to reduce computational complexity of a filtering algorithm for its use in real-time applications.

Hence, it may be concluded that there is enough scope to develop better filtering schemes with very low computational complexity that may yield high noise reduction as well as preservation of edges and fine details in an image.

1.4 The Problem Statement

It is essential to suppress noise from an image as far as possible. At the same time, its fine-details and edges are to be retained as much as practicable. The filtering

algorithms to be developed must be of low computational complexity so that they can filter noise in short time, and hence will find themselves suitable for online and real-time applications.

Thus, the problem taken for this doctoral research work is to develop *efficient non-linear filters to suppress impulse noise*:

- *with very high efficiency*
- *yielding extremely low distortion*
- *in wide range of noise densities*
- *with less computational complexity and low run-time overhead*
- *while retaining edges and fine details of an image*

This research work focuses mainly on salt-and-pepper impulse noise; in addition, some methods are developed to suppress both random-valued and salt-and-pepper impulse noise.

Usually, transform-domain filters consume much more time compared to the time taken by spatial-domain filers. Thus it is intended to develop efficient filters only in spatial-domain.

Therefore, the following problem is taken.

Problem: *To develop some novel efficient restoration algorithms for images corrupted with high density impulse noise.*

A brief overview of fundamentals of spatial-domain filtering is presented in the next section for ready reference.

1.5 Basics of Spatial-Domain Filtering

Let $f(i,j)$ represent an original noise free digital image with M-rows and N-columns with the spatial indices *i* and *j* ranging from 0 to M-1 and 0 to N-1 respectively. It is denoted as:

$$f(i,j) = \begin{bmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \dots & \vdots \\ f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Let $g(i,j)$ represent the noisy image with same dimension as that of $f(i,j)$.

Let us define $\mathrm{W}(k,l)$ as a mask or window or kernel, $k,l \in \mathbb{Z}$, k and l are limited in the range of $\frac{-(M_w-1)}{2} \leq k \leq \frac{(M_w-1)}{2}$ and $\frac{-(N_w-1)}{2} \leq l \leq \frac{(N_w-1)}{2}$, where M_w and N_w represent the number of rows and columns in the window. For example if it is (3×3)

then, the range of k and l is given by $-1 \leq k \leq +1$ and $-1 \leq l \leq +1$ respectively. A noisy sub-image $g_{k,l}(i,j)$ for (3×3) with $g(i,j)$ as a centre pixel is given by:

$g_{k,l}(i,j) = g(i+k, j+l)$ for $-1 \leq (k,l) \leq +1$. It is usually expressed, in matrix form, as:

$$g_{k,l}(i,j) = \begin{bmatrix} g(i-1,j-1) & g(i-1,j) & g(i-1,j+1) \\ g(i,j-1) & g(i,j) & g(i,j+1) \\ g(i+1,j-1) & g(i+1,j) & g(i+1,j+1) \end{bmatrix}$$

Similarly, a (5×5) sub-image centered at $g(i,j)$ is given by:

$$g_{k,l}(i,j) = g(i+k, j+l), \quad -2 \leq (k,l) \leq +2.$$

The filtering process consists simply of moving the filtering mask from point to point in the image. At each point (i,j), the response of the filter at that point is calculated using predefined relationships. For example, if it is mean filter, then, the centre pixel is replaced by mean value of pixels in the filtering window, if it is median filtering, centre pixel is replaced by median of sub-image pixels.

Thus, a restored image is evaluated by convolving the noisy image $g(i,j)$ with filter kernel $W(k,l)$. The convolution process is mathematically represented as:

$$\hat{f}(i,j) = \sum_{l} \sum_{k} g_{k,l}(i,j) \times W(k,l)$$

where, $\hat{f}(i,j)$ denotes the restored image.

1.6 Image Metrics

The performances of filters are evaluated by objective as well as subjective techniques. For subjective evaluation, the image has to be observed by a human expert [168] whereas objective evaluation of an image is performed by evaluating error and error-related parameters mathematically.

There are various metrics used for objective evaluation of an image. The commonly used metrics are mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and peak signal to noise ratio (PSNR) etc. [6,169].

The original noise-free image, noisy image, and the filtered image are represented by $f(i,j)$, $g(i,j)$ and $\hat{f}(i,j)$ respectively. Let the images be of size $M \times N$, i.e. $i=1,2,3,\ldots,M$, and $j=1,2,3,\ldots,N$. Then, MSE is defined as:

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(\hat{f}(i,j) - f(i,j))^2}{M \times N} \tag{1.7}$$

The PSNR is defined in logarithmic scale, and is expressed in dB. It is a ratio of peak signal power to noise power. The PSNR is defined as:

$$PSNR = 10.\log_{10}(\frac{1}{MSE}) \text{ dB} \tag{1.8}$$

provided the signal lies in the range [0,1]. On the other hand, if the signal is represented in the range of [0,255], the numerator in (1.8) will be $(255)^2$ instead of 1.

For the color image processing, the color peak signal to noise ratio (CPSNR) [36b] in dB is used as performance measure. The CPSNR is defined as:

$$CPSNR = 10 \cdot \log_{10}\left[\frac{1}{3}\sum_{c\in R,G,B} MSE_c\right]^{-1} \text{dB} \tag{1.9}$$

where, MSE_c is the mean squared error in a particular channel of the color space.

Though these image metrics are extensively used for evaluating the quality of a restored image, none of them gives a true indication of performance of a filter. In addition to these parameters, a new metric: universal quality index (UQI) [170] is used in literature to evaluate the quality of an image.

Universal Quality Index:

The universal quality index (UQI) is modeled by considering three different factors: (i) loss of correlation, (ii) luminance distortion and (iii) contrast distortion. It is defined by:

$$UQI = \frac{\sigma_{f\hat{f}}}{\sigma_f \sigma_{\hat{f}}} \cdot \frac{2\bar{f}\,\bar{\hat{f}}}{\bar{f}^{\,2} + \bar{\hat{f}}^{\,2}} \cdot \frac{2\sigma_f \sigma_{\hat{f}}}{\sigma_f^2 + \sigma_{\hat{f}}^2} \tag{1.10}$$

where,

$$\bar{f} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j) \tag{1.11}$$

$$\bar{\hat{f}} = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} \hat{f}(i,j) \tag{1.12}$$

$$\sigma_f^2 = \frac{1}{MN-1}\sum_{i=1}^{M}\sum_{j=1}^{N}(f(i,j) - \bar{f}(i,j))^2 \tag{1.13}$$

$$\sigma_{\hat{f}}^2 = \frac{1}{MN-1}\sum_{i=1}^{M}\sum_{j=1}^{N}(\hat{f}(i,j) - \bar{\hat{f}}(i,j))^2 \tag{1.14}$$

$$\sigma_{f\hat{f}} = \frac{1}{MN-1}\sum_{i=1}^{M}\sum_{j=1}^{N}(f(i,j) - \bar{f}(i,j))(\hat{f}(i,j) - \bar{\hat{f}}(i,j)) \tag{1.15}$$

The UQI consists of three components. The first component is the correlation coefficient between the original noise-free image, f and the restored image, $\hat{f}$ that measures the degree of linear correlation between them, and its dynamic range is [-1,1]. The second component, with a range of [0, 1], measures the closeness between the average luminance of f and $\hat{f}$. It reaches the maximum value of 1 if and only if $\bar{f}$ equals $\bar{\hat{f}}$. The standard deviations of these two images, σ_f and $\sigma_{\hat{f}}$ are also regarded as estimates of their contrast levels. The value of contrast level ranges from 0 to 1 and the optimum value of 1 is achieved only when $\sigma_f = \sigma_{\hat{f}}$.

Hence, combining the three parameters: correlation, average luminance similarity and contrast-level similarity, the new image metric: universal quality index (UQI) becomes a very good performance measure.

Image Enhancement Factor:

The next most widely used quality metric for image quality measurement is Image Enhancement Factor (IEF) [171]. It indicates the performance of a filter under varying noise densities. Thus, IEF indicates qualitatively the relative quality improvement (noise-reduction) exhibited by a process (filter). The mathematical representation of IEF is given by,

$$IEF = \frac{\sum[g(i,j)-f(i,j)]^2}{\sum\left[\hat{f}(i,j)-f(i,j)\right]^2} \tag{1.16}$$

The above metrics are extensively used to evaluate the restored image quality of filter, and none of them gives the indication of complexity of filter. Hence, another parameter, execution time, is employed to measure the complexity of filter.

Execution Time:

Execution Time (T_E) of a filter is defined as *the time taken by a digital computing platform to execute the algorithm, when no other software except the operating system (OS) runs on it*.

Execution Time (T_E) depends on the configuration of computer used for execution of algorithm. Based of complexity of filter the execution time varies. The filter with less complexity will take less time. The filter with low *execution time* is preferred for online and real-time applications.

Hence, a filter with lower T_E is better than a filter having higher T_E value when all other performance-measures are identical.

Since the execution time is platform dependant, some standard hardware computing platforms: SYSTEM-1, SYSTEM-2 and SYSTEM-3 presented in Table-1.1 are taken for the simulation work. Thus, the T_E parameter values for the various existing and proposed filters are evaluated by running these filtering algorithms on these platforms.

Table-1.1: Details of hardware platforms (along with their operating system) used for simulating the filters

Hardware platforms	Processor	Clock (GHz)	RAM (GB) (usable)	Operating System (OS)
SYSTEM-1	Pentium (R)D Processor	2.80	0.99	Windows XP 32 bit OS
SYSTEM-2	Intel(R),Core(TM) 2Duo	3	3.4	Windows XP 32 bit OS
SYSTEM-3	Intel(R),Core(TM) i5	3.2	3.4	Windows XP 32 bit OS

1.7 Chapter-wise Organization of the Thesis

The chapter-wise organization of the thesis is given below.

Chapter-1: Introduction

Chapter-2: Study of Existing Filters

Chapter-3: Development of Novel Filters for Suppression of Salt-and-Pepper Noise

Chapter-4: Development of Novel Filters for Suppression of Random-Valued Impulse Noise

Chapter-5: Development of Some Color Image Denoising Filters

Chapter-6: Conclusion

1.8 Conclusion

In this chapter, the basics of Digital Image Processing, sources of noise and different types of noise, review of some existing methods and some commonly used image metrics for performance measure of filters are discussed. After brief literature review, the doctoral research problem is evolved.

Extensive studies of well known and high-performing image denoising filters available in literature are presented in the next chapter whereas the proposed algorithms are discussed in subsequent chapters. Finally, the dissertation is concluded in Chapter-6.

Chapter 2

Study of Image Denoising Filters

2

Preview

Image noise suppression is a highly demanded approach in digital imaging systems design. Impulsive noise is frequently encountered during the processes of acquisition, transmission and storage and retrieval. In the area of image denoising many filters are proposed in literature. The main steps in this process are classification (detection) and reconstruction (filtering). Classification is used to separate uncorrupted pixels from corrupted pixels. Reconstruction involves replacing the corrupted pixels by an estimation technique.

There are various filters existing in literature, which are used for filtering salt-and-pepper impulse noise and random-valued impulse noise. There are some special types of filters which are used for suppressing salt-and-pepper noise as well as random-valued impulse noise. In this chapter, some well-known, standard and benchmark filters, which are available in literature, are studied. Novel filters, designed and developed in this research work, are compared against these filters in subsequent chapters. Therefore, attempts are made here for detailed and critical analysis of these existing filters.

The organization of the chapter is given below.

- Order Statistics Filters
- Detection Followed By Filtering
- A Brief Comparative Performance Analysis
- Conclusion

2.1 Order Statistics Filters

Order statistic filters are non-linear spatial filters whose response is based on ordering (ranking) the pixels contained in the area encompassed by the filtering window. Usually, sliding window technique [1, 2, 6] is employed to perform pixel-by-pixel operation in a filtering algorithm. The local statistics obtained from the neighborhood of the center pixel give a lot of information about its expected value. If the neighborhood data are ordered (sorted), then ordered statistical information is obtained. The center pixel in the sliding window is replaced with the value determined by the ranking result.

For example, if a 3×3 window is used for spatial sampling, then 9 pixel data are available at a time. First of all, the 2-D data is converted to a 1-D data, i.e. a vector. Let this vector of 9 data be sorted. Then, if the mid value (5^{th} position pixel value in the sorted vector of length = 9) is taken, it becomes **median** filtering with the filter weight vector [0 0 0 0 1 0 0 0 0]. The **median**, **alpha-trimmed mean** (ATM), **min**, **max** filters are some members of this interesting family.

2.1.1 Median Filter

The median filter is one of the most popular nonlinear filters [1, 2]. It is very simple to implement and much efficient as well. The median filter, especially with larger window size destroys the fine image details due to its rank ordering process. It acts like a low pass filter which blocks all high frequency components of the image like edges and noise, thus blurs the image.

As the noise density increases, the filtering window size is increased to have sufficient number of uncorrupted pixels in the neighborhood. Depending upon the sliding window mask, there may be many variations of median filters. In this thesis, Median filter with sliding window (3×3), (5×5) and (7×7) are reviewed. A centre pixel, irrespective of either being noisy or not, is replaced with the median value. Due

to this, its results are disappointing in many cases. Applications of the median filter require caution because median filtering tends to remove image details such as thin lines and corners while reducing noise.

2.1.2 Alpha-Trimmed Mean Filter

The alpha-trimmed mean (ATM) filter [67] is based on order statistics and varies between a median and mean filter. It is so named because, rather than averaging the entire data set, a few data points are removed (trimmed) and the remainders are averaged. The points which are removed are most extreme values, both low and high, with an equal number of points dropped at each end (symmetric trimming). In practice, the alpha-trimmed mean is computed by sorting the data low to high and finding the average of the central part of the ordered array. The number of data values which are dropped from the average is controlled by trimming parameter $\propto$ (alpha) and hence the name alpha-trimmed mean filter.

Let $g_{k.l}(i,j)$be a sub-image of noisy image $g(i,j)$. For simplicity, $g_{k.l}(i,j)$ is referred as $g_{k.l}$. Suppose the $\frac{\propto}{2}$ lowest and the $\frac{\propto}{2}$ highest gray-level values of $g_{k.l}$ are deleted from the neighborhood. Let g_r represent the remaining $(mn-\propto)$ pixels. A filter formed by averaging these remaining pixels is called alpha- trimmed mean filter whose output may be expressed as:

$$\hat{f}(i,j) = \frac{1}{mn-\alpha}\sum g_r \qquad (2.1)$$

Choice of parameter $\propto$ is very critical and it determines the filtering performance. Hence, the ATM filter is usually employed as an adaptive filter whose $\propto$ may be varied depending on the local signal statistics. Therefore, it is a computation-intensive filter as compared to a simple median filter. Another problem of ATM is that the detailed behavior of the signal cannot be preserved when the filter window is large.

2.1.3 Center Weighted Median Filter (CWM)

The center weighted median (CWM) [64] filter is a special case of weighted median (WM) filters. This filter gives more weight only to the central pixel of a window and thus it is easy to design and implement. CWM filter preserves more details at the expense of less noise suppression like other non-adaptive detail preserving filters.

Let $g(i,j)$ be a noisy image. Consider a sub-image $g_{k,l}(i,j)$ of size P = Q = 2L+1, centered at (i,j). The output of the CWM filter, in which a weight adjustment is applied to the centre pixel within the sliding window, can be described as

$$\hat{f}(i,j) = med(\{g_{k,l}(i-k,j-l) | (k,l) \neq (0,0), w_c copies of g_{k,l}(i,j) | (k,l) = (0,0)\}) \quad (2.2)$$

For a (3×3) window, the median is computed based on those 8+ w_c pixel values.

Note that integer w_c is positive and odd, and the CWM filter becomes the median filter when w_c =1. On the other hand, when w_c is greater than or equal to the window size (e.g.,$w_c \geq 9$ for a (3×3) window), it becomes an *identity filter*, which always takes the origin pixel value$g(i,j)$as the output. A CWM filter with a large center weight performs better in detail preservation. But its performance is not acceptable at high noise densities.

2.2 Detection Followed by Filtering

The filters which are discussed in section 2.1 are the filters without noise detection stage. Thus, even non-noisy pixels are also replaced by some estimator. Because of this, the performance of these filters is not good. To overcome this problem, a new filtering technique is introduced. This type of filtering involves two steps. In first step it identifies noisy pixels and in second step it filters only those pixels that are identified as noisy. The performance of these filters depends on impulse detector and estimator by which noisy pixels are replaced in the filtering process.

In this section some well-known, standard and benchmark filters, available in literature, are studied.

2.2.1 Tri-State Median Filtering (TSM)

The tri-state median (TSM) filter [86] incorporates the median filter (MF) and the center weighted median (CWM) filter in a noise detection framework. Noise detection is realized by an impulse detector, which takes the outputs from the median and center weighted median filters and compares them with the center pixel value in order to make a tri-state decision. The switching logic is controlled by a threshold value. Depending on this threshold value, the center pixel value is replaced by the output of

either median filter (MF), CWM filter or identity filter. The output $\hat{f}(i,j)$ of TSM is given by

$$\hat{f}(i,j) = \begin{cases} g(i,j), & T \le d_1 \\ g^{CWM}(i,j), & d_1 < T \le d_2 \\ g^{MF}(i,j), & T > d_2 \end{cases} \quad (2.3)$$

where, $g^{CWM}(i,j)$ and $g^{MF}(i,j)$ are the outputs of CWM and MF filters respectively, $g(i,j)$ is noisy image and $d_1 = \left|g(i,j) - g^{MF}(i,j)\right|$ and $d_2 = \left|g(i,j) - g^{CWM}(i,j)\right|$. Note that the threshold T affects the performance of impulse detection. Usually, a threshold, T ∈ [10, 30] is good enough [86]. Of course, its value should adaptively be chosen for better results.

2.2.2 Adaptive Median Filters (AMF) [75]

For good impulse classification it is preferred to remove the positive and negative impulse noise one after another. There are a number of algorithms which resolve this problem, but they are more complex. This algorithm is simple and better performing in removing a high density of impulse noise as well as non-impulse noise while preserving fine details. The size of filtering window of median filter is adjusted based on noise density.

This algorithm is based on two level tests. In the first level of tests, the presence of residual impulse in a median filtered output is tested. If there is no impulse in the median filtered output, then the second level tests are carried out to check whether the center pixel itself is corrupted or not. If the center pixel is uncorrupted then it is retained at the output of filtered image. If not, the output pixel is replaced by the median filter output. On the other hand, if the first level detects an impulse, then the window size for median filter is increased and the first level tests are repeated. The maximum filtering window size taken is 11×11 if the noise density is of the order of 70% [75].

2.2.3 Progressive Switching Median (PSM) Filter for the Removal of Impulse Noise from Highly Corrupted Images

Progressive switching median (PSM) filter is median based filter [72]. It consists of two points (i) switching scheme – an impulse detection algorithm is used before filtering; thus only noisy pixels are filtered and (ii) progressive methods – both

impulse detection and progressive filtering are applied through several iterations one after the other. Hence, it is referred as PSM filter.

In the first stage, an impulse detection algorithm is used to generate a sequence of binary flag images. This flag image indicates the location of noise in the input image. If the binary flag image pixel is 1, it indicates that the pixel in that position in the input image is noisy. On the other hand, if the binary flag is 0, then it is considered noise-free. In the second stage, filtering is applied based on binary flag image generated in the first stage. Both these steps are progressively applied through several iterations. The noisy pixels processed in the current iteration are used to help the process of the other pixels in the subsequent iterations. Therefore, better restoration results are expected, even under high noise density conditions.

2.2.4 A New Impulse Detector for Switching Median Filter (SMF) [73]

An impulse detector which is proposed for switching median filter is based on the minimum absolute value of four convolutions obtained using one-dimensional Laplacian operators.

The input image is first convolved with a set of convolution kernels. Here, four one-dimensional Laplacian operators as shown in Fig 2.1 are used, each of which is sensitive to edges in a different orientation. Then, the minimum absolute value of these four convolutions is used for impulse detection, which can be represented as:

$$r(i,j) = min\left(\left\langle \left| g(i,j) \otimes K_p \right| \quad \middle| \, p = 1,2, \ldots, 4 \right\rangle\right) \tag{2.4}$$

where K_p is the p[th] kernel and the symbol, $\otimes$, denotes a convolution operation.

The value of $r(i,j)$ detects impulses due to the following reasons.

(1) $r(i,j)$is large when the current pixel is an isolated impulse because the four convolutions are large and almost the same.

(2) $r(i,j)$ is small when the current pixel is a noise-free flat region pixel because the four convolutions are close to zero.

(3) $r(i,j)$ is small even when the current pixel is an edge (including thin line) pixel because one of the convolutions is very small (close to zero) although the other three might be large.

From the above analysis, it is evident that $r(i,j)$ is large when $g(i,j)$ is corrupted with an impulsive noise, and $r(i,j)$ is small when $g(i,j)$ is noise-free

whether or not it is a flat-region, edge, or thin-line pixel. So, the $r(i,j)$ is compared with a threshold to determine whether a pixel is corrupted or uncorrupted. The binary flag image $b(i,j)$ is given by,

$$b(i,j) = \begin{cases} 1, & r(i,j) > T \\ 0, & r(i,j) \leq T \end{cases} \tag{2.5}$$

The filtered image is given by

$$\hat{f}(i,j) = b(i,j) \times m(i,j) + (1 - b(i,j)) \times g(i,j) \tag{2.6}$$

where $m(i,j)$ is median value of filtering window. Based on the number of simulations carried out on different test images, the threshold, T ∈ [30, 50] [73]. The algorithm is tested with a threshold, T=40 and filtering window of size 5 × 5.

0	0	0	0	0
0	0	0	0	0
-1	-1	4	-1	-1
0	0	0	0	0
0	0	0	0	0

0	0	-1	0	0
0	0	-1	0	0
0	0	4	0	0
0	0	-1	0	0
0	0	-1	0	0

-1	0	0	0	0
0	-1	0	0	0
0	0	4	0	0
0	0	0	-1	0
0	0	0	0	-1

0	0	0	0	-1
0	0	0	-1	0
0	0	4	0	0
0	-1	0	0	0
-1	0	0	0	0

Fig.2.1. Four 5×5 convolution kernels

2.2.5 Advanced Impulse Detection Based on Pixel-Wise MAD (PWMAD) [112]

This method is used for filtering both *random valued* and *salt-and-pepper valued* impulse noise. In this method, median of the absolute deviations from the median, MAD [112] is modified and used to efficiently separate noisy pixels from the image details. An iterative pixel-wise modification of MAD, PWMAD provides reliable removal of arbitrarily distributed impulse noise.

Let $g(i,j)$, $m(i,j)$and $d(i,j)$represent pixels with coordinates (*i*, *j*) of noisy image, median image and absolute deviation image, respectively. Also, let **g**(*i*,*j*), **m**(*i*,*j*)and **d**(*i*,*j*)denote matrices (sub-image) whose elements are pixels of the corresponding images contained within the (2*L* + 1) × (2*L* + 1) size window, centered

around at position (*i, j*). The median image and absolute deviation image may be defined as:

$$m(i,j)=med(\mathbf{g}(i,j)) \tag{2.7}$$

$$d(i,j)= |\, g(i,j)- m(i,j)| \tag{2.8}$$

The median of the absolute deviations from the median, MAD, image is defined as:

$$MAD(i,j)= med\,(|\mathbf{g}(i,j)- med\,(g(i,j))|) \tag{2.9}$$

Note that a single median value is subtracted from all the pixels within $\mathbf{g_n}(i, j)$. In order to make MAD consistent with definition of absolute deviation image, where its corresponding median image pixel $m(i,j)$ is subtracted from each pixel, a modified Pixel-Wise MAD (PWMAD) image is given by

$$PWMAD\,(i,j)= med\,(d(i,j)) =med\,(|\mathbf{g}\,(i,j)-\mathbf{m}(i,j)|) \tag{2.10}$$

The absolute deviation image $d(i,j)$consists of noise and image details eliminated from the *noisy image* by *median filtered*. If a median is applied to $d(i,j)$ (absolute deviation image), a PWMAD image is generated. By subtracting the PWMAD image from $d(i,j)$, details are eliminated and only noise is left behind. If this process is repeated several times, then the image, obtained after the final iteration, consists of pixels that are corrupted with impulsive noise. This image can be used for generation of binary image.

The whole iteration procedure can be represented as:

$$d^{(\mathrm{n}+1)}(\mathrm{i,j}) = \left|d^{(\mathrm{n})}(\mathrm{i,j}) - \mathrm{PWMAD}\,(d^{(n)}(\mathrm{i,j}))\right|$$

$$\text{i.e.}\quad d^{(\mathrm{n}+1)}(\mathrm{i,j}) = \left|d^{(\mathrm{n})}(\mathrm{i,j}) - med\,(\mathbf{d}^{(n)}(\mathrm{i,j}))\right| \tag{2.11}$$

where $d^{(0)}(\mathrm{i,j})$ is a primary absolute deviation image defined in (2.8). The iteration is terminated after n = N-1, and $d^{(\mathrm{N})}(\mathrm{i,j})$, thus obtained, is used for generation of binary flag image, which is defined as

$$b(i,j) = \begin{cases} 1, & d^{(\mathrm{N})}(i,j) \geq T \\ 0, & d^{(\mathrm{N})}(i,j) < T \end{cases} \tag{2.12}$$

The value of *T* is in the range [0 30]. The simulation is carried with T = 5 and number of iterations, N = 3 and the results are presented in the Chapter-3.

The output image is given by (2.6), i.e. selective median filtering is performed.

2.2.6 Impulse Noise Filter with Adaptive MAD (AMAD)-Based Threshold [129]

This is an improved method of PWMAD. This is also used for filtering both *random valued* and *salt-and-pepper valued* impulse noise. In this method, an extension to the switching scheme is used, where the threshold T is varying from pixel to pixel. The threshold value is modified in accordance with variance, estimated by using MAD. No iteration is used for impulse detection, which reduces run time with same quality as compared to PWMAD. The threshold T is given by

$$T = \begin{cases} a, & MAD(i,j) \geq b \\ a + \frac{b-a}{a} \times MAD(i,j), & MAD(i,j) < b \end{cases} \quad (2.13)$$

where a and b are varying parameters, $a \in [10, 30]$; $b \in [50, 100]$ [129].The simulation is carried by taking a=15 and b=70, and the results are presented in Chapter-4.

2.2.7 A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images [106]

To determine whether a pixel is corrupted or not, the Boundary Discriminative Noise Detection (BDND) algorithm [106] first classifies the pixels of a localized window, centering on the current pixel, into three groups: *lower intensity impulse noise*, *uncorrupted pixels*, and *higher intensity impulse noise*. The center pixel will then be considered as uncorrupted, provided that it belongs to the *uncorrupted* pixel group, else it is considered *corrupted*. The grouping of pixels depends on two boundaries. The accurate determination of these boundaries yields very high noise detection accuracy even up to 70% noise corruption.

The algorithm is applied to each pixel of the noisy image in order to identify whether the pixel is *corrupted* or *uncorrupted*. After such an application to the entire image, a binary decision map $b(i,j)$ is formed with **0**s indicating the positions of *uncorrupted* pixels (i.e., $b(i,j) = 0$), and **1s** for those *corrupted* ones (i.e., $b(i,j) = 1$). To accomplish this objective, all the pixels within a pre-defined window that centers at the considered pixel are grouped into three clusters, *low-intensity cluster*, *medium-intensity cluster* and *high-intensity cluster*. For each pixel $g(i,j)$ being considered, if $0 \leq g(i,j) \leq b_1$, the pixel will be assigned to the *lower*-intensity cluster; otherwise, to the *medium*-intensity cluster for $b_1 < g(i,j) \leq b_2$ or to the

high-intensity cluster for $b_2 < g(i,j) \leq 255$. The current pixel is identified as *uncorrupted* only if it falls into the medium-intensity cluster; otherwise it is classified as corrupted.

The *boundary discriminative* process consists of two iterations in which the second iteration will only be invoked conditionally. In the first iteration, a local window with a size of 21×21 is used to examine whether the center pixel is an uncorrupted one [106]. If the pixel fails to meet the condition to be classified as uncorrupted, the second iteration will be invoked to further examine the pixel based on a more confined local statistics by using a 3×3 window. In summary, the steps of the BDND are:

Step-1. A sliding window of size 21×21 is centered around the current pixel.

Step-2. Sort the pixels in the window according to the ascending order and find the median, $m(i,j)$, of the sorted vector V_o.

Step-3. Compute the intensity difference between each pair of adjacent pixels across the sorted vector V_o and a difference vector V_d is obtained.

Step-4. For the pixel intensities between 0 and *med* in the V_o, find the maximum intensity difference in the V_d of the same range and mark its corresponding pixel in the V_o as the boundary b_1 .

Step-5. Likewise, the boundary b_2 is identified for pixel intensities between *med* and 255; three clusters are, thus, formed.

Step-6. If the pixel belongs to the middle cluster, it is classified as *uncorrupted* pixel, and the classification process stops; else, the second iteration will be invoked which is given by step-7 and step-8.

Step-7. Impose a 3×3 window, being centered around the concerned pixel and repeat the steps: Step-2 through Step-5.

Step -8. If the pixel under consideration belongs to the middle cluster, it is classified as *uncorrupted* pixel; otherwise, *corrupted*.

Adaptive Filtering Scheme:

In the filtering process a binary flag image is used. The pixel which is declared as noisy (i.e.$b(i,j) = 1$), is replaced with median of uncorrupted pixels in the filtering window. If the pixel is noise-free (i.e.$b(i,j) = 0$), it is retained in the reconstructed

image $\hat{f}(i,j)$. Thus, it passes through a selective median filtering process and hence the output image is represented by (2.6).

Now, $m(i,j)$ is median of only the uncorrupted pixels in the adaptive window, $\{g_{k,l}\}$. Starting the filtering process with W=3×3, the filtering window iteratively extends outward by one pixel in all the four sides of the window, provided that the number of uncorrupted pixels are less than half of the total number of pixels within the filtering window, while $W<W_D$ or number of uncorrupted pixels is equal to zero, W_D is maximum filtering window size. In this work, an additional reliability condition is further imposed such that the filtering window will also be extended when the number of uncorrupted pixels is equal to zero.

The performances of all the above algorithms are tested with different gray scale images, with their dynamic range of values (0, 255). In each simulation, image is corrupted by impulse noise with equal probability at different noise densities. The restoration performances are quantitatively measured by using different image metrics like PSNR, MSE, IEF and UQI. All the simulation results are presented in next chapters. The salt-and-pepper noise related filters are analysed in Chapter-3 and random-valued impulse noise filters are analysed in Chapter-4.

Though the detail performances of these filters are presented in subsequent chapters, a brief comparative performance analysis is presented below for ready reference.

2.3 A Brief Comparative Performance Analysis

A brief comparative performance analysis is presented, in terms of PSNR, as a ready reference. The existing well known filters are simulated on MATLAB 7.4 platform. The *Lena* image of size 512×512, an 8 bit gray-scale image, is employed as test image. The input image is corrupted with salt-and-pepper and random-value impulse noise with noise density ranging from 10% to 90% and 5% to 30% respectively. The peak-signal-to-noise ratio (PSNR) is used as performance measure. The highest (best) PSNR value for a particular noise density is highlighted to show the best performance.

From Table 2.1, it is observed that the filter BDND performs better in terms of PSNR in complete range of noise density. Still, the no filter shows the best perform in the range of 50% to 90%. The filter PSM and simple MF [3×3] perform second best, but fail to perform well under high noise density. ATM [7×7] gives second best performance in medium range (50% to 70%) of noise density.

Table-2.1: Filtering performance of various filters in terms of PSNR (dB)
Test image: Lena

Sl.No	Filters	% of Noise (Salt-and-Pepper)					
		10	**20**	**30**	**50**	**70**	**90**
1	MF [3×3]	33.74	27.28	21.75	14.12	9.6	6.54
2	MF [5×5]	31.44	30.67	29.11	20.32	12.88	7.29
3	MF [7×7]	29.41	28.99	28.49	25.91	16.54	8.09
4	ATM[3×3]	31.99	27.72	23.06	15.97	11.31	8.45
5	ATM[5×5]	29.11	27.61	25.96	20.68	14.57	10.92
6	ATM[7×7]	29.2	28.88	28.46	**27.11**	**18.92**	8.62
7	CWM	34.49	30.11	24.01	15.55	10.03	6.69
8	TSM	35.49	29.31	23.48	15.02	9.58	6.27
9	AMF	33.76	29.51	24.65	16.37	10.87	6.92
10	PSM	**37.27**	**32.81**	**29.21**	9.95	8.11	6.61
11	SMF	29.24	27.51	25.94	21.12	13.38	7.36
12	BDND	**39.09**	**36.53**	**34.22**	**29.66**	**25.62**	**16.81**
Sl.No	**Filters**	**% of Noise (Random-valued Impulse Noise)**					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	33.95	29.86	24.56	20.57	17.31	14.85
2	ATM[3×3]	33.66	30.02	25.81	22.22	18.86	16.15
3	TSM	34.86	30.10	24.23	21.22	17.63	15.10
4	PSM	34.93	31.55	28.13	24.19	20.89	17.36
5	PWMAD	33.52	28.28	22.28	19.01	15.84	15.84
6	**AMAD**	**36.01**	**31.94**	**27.01**	**24.22**	**21.05**	**17.56**

From second part of Table 2.1, it is evident that AMAD performs best at low density of random noise. Even MF [3×3], PSM and ATM are also exhibiting reasonably good performance in this range of noise density. But all filters fail to perform in high range of RVIN.

2.4 Conclusion

This chapter aims to provide a complete scenario of some existing filters. Because of space limit, only a few important filters are presented in this chapter.

From Table-2.1 it is observed that the BDND filter performs best for SPN of low, medium and high noise densities. The performances of other filters are restricted to either low range (i.e., 10% to 30%) or medium range (i.e., 30% to 50%). The performances of filters available in literature for RVIN are also observed in this table. These filters don't exhibit any promising results. Some of them perform well at low noise density whereas some other show better results at medium or high noise densities.

Hence, there is sufficient scope to develop more efficient filters to suppress SPN and RVIN of wide noise densities. The filters, whose performances are studied through Table-2.1, will be employed as references in subsequent chapters where new filters developed will be compared against them.

Chapter 3

Development of Novel Filters for Suppression of Salt-and-Pepper Noise

3

Preview

In this chapter some new filters for suppressing salt-and-pepper impulse noise are proposed that works based on decision-based techniques. The simulation results, presented at the end of the chapter, are quite encouraging. The developed efficient spatial-domain image denoising algorithms that are presented here are:

- *Adaptive Noise Detection and Suppression* (***ANDS)*** *Filter*
- *Robust Estimator based Impulse-Noise Reduction* (***REIR***) *Algorithm*
- *Impulse Denoising Using Improved Progressive Switching Median Filter* (***IDPSM***)
- *Impulse-Noise Removal by Impulse Classification* (***IRIC***)
- *A Novel Adaptive Switching Filter-I* (***ASF-I***) *for Suppression of High Density SPN*
- *A Novel Adaptive Switching Filter-II* (***ASF-II***) *for Suppression of High Density SPN*
- *Impulse Denoising Using Iterative Adaptive Switching Filter* (***IASF***)

Before describing the newly developed filters, Basic Filter Paradigms are discussed in the next section.

3.1. Basic Filter Paradigms

The filters, developed in this doctoral research work, are basically decision-directed filters. Decision directed filters are also known as Classifier-Filter (CF) algorithms since the input data is first classified as either noisy or noise-free and then filtering operation is performed only if the input data has been classified noisy. The CFs come under four basic paradigms shown in Fig. 3.1.The earliest in the history is the switching filter (SF) paradigm, depicted in Fig. 3.1(a), whereas the basic classifier filter (BCF), shown in Fig. 3.1(b), is a slight modified version of it. In the BCF framework, an algorithm needs to develop a binary flag image, $b(i,j)$. On the other hand, an SF paradigm doesn't produce any such intermediate image and thus the classification and the filtering operation are concurrently performed. The third and the fourth paradigms: iterative classifier filters, namely, ICF-1 and ICF-2, perform the classification iteratively. While ICF-1 doesn't employ adaptive windowing, ICF-2 does employ for much better classification at very high noise densities. The ICF-1 and ICF-2 paradigms are illustrated in Fig. 3.1(c) and Fig. 3.1(d) respectively. Novel spatial domain filters are developed, in this research work, on the last three basic frameworks, namely, BCF, ICF-1 and ICF-2. The proposed algorithms and the underlying paradigms are listed in the Table-3.1.

Table-3.1: Proposed algorithms with basic paradigms

Sl. No.	Algorithm	Paradigm
1	Adaptive Noise Detection and Suppression (**ANDS**) Filter	BCF
2	Robust Estimator Based Impulse-noise Reduction Algorithm (**REIRA**)	ICF-2
3	Impulse Denoising Using Improved Progressive Switching Median Filter (**IDPSM**)	ICF-1
4	Impulse Noise Removal in Highly Corrupted Image by Impulse Classification (**IRIC**)	SF
5	A Novel Adaptive Switching Filter-I (**ASF-I**) for suppression of High Density SPN	BCF
6	A Novel Adaptive Switching Filter-II (**ASF-II**) for suppression of High Density SPN	BCF
7	Impulse Denoising Using Iterative Adaptive Switching Filter (**IASF**)	ICF-1

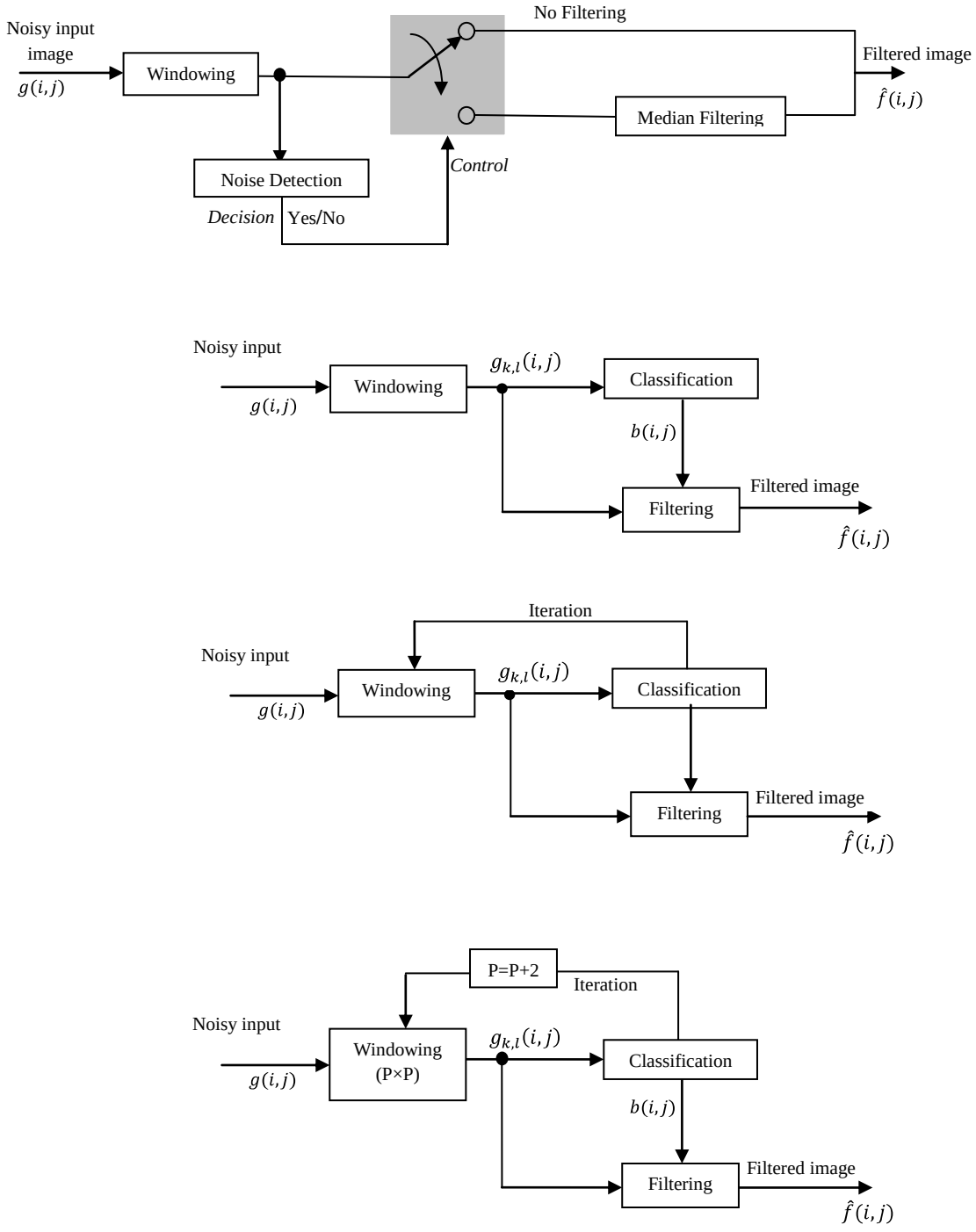


Fig. 3.1 Block Diagrams of Basic Paradigms

a **(a) Switching-Filter (SF)**

b **(b) Basic Classifier-Filter (BCF)**

c **(c) Iterative Classifier-Filter-1 (ICF-1)**

d **(d) Iterative Classifier-Filter-2 (ICF-2)**

3.2. Adaptive Noise Detection and Suppression (ANDS) Filter [P1]

This method is based on the BCF paradigm shown in Fig. 3.1(b). Neighborhood difference is employed for pixel classification. Controlled by binary image, $b(i,j)$, the noise is filtered by estimating the value of a pixel with an adaptive switching based median filter applied exclusively to neighborhood pixels that are labeled noise-free. The proposed filter performs better in retaining edges and fine details of an image at low-to-medium densities of fixed-valued impulse noise.

3.2.1 Adaptive Noise Detection Algorithm

Fig. 3.2 shows the flowchart for noise detection algorithm. The following steps explain the noise detection algorithm.

Step-1. Neighborhood Preprocessing.

A 3×3 window of the noisy input image is taken around a pixel $g(i,j)$ that is, $g_{k,l}(i,j) = g(i+k, j+l)$ for$-1 \leq (k,l) \leq +1$. The sub-image$g_{k,l}(i,j)$, $D_{k,l}(i,j)$ and $M_{k,l}(i,j)$ are denoted as $g_{k,l}$, $D_{k,l}$ and $M_{k,l}$ respectively. The difference image, $D_{k,l}(i,j)$ is then evaluated as:

$$D_{k,l}(i,j) = g(i,j) - g_{k,l}$$

Step 2: Neighborhood Replacement

Replace all neighboring pixels with the corresponding difference values, i.e.

$$g_{k,l}(i,j) = D_{k,l}(i,j) \forall k \neq 0, \quad l \neq 0.$$

Step-3. Correlation Map using Adaptive Thresholding

In this step correlation map to eight neighborhood of $D_{k,l}(2,2)$ is developed. Mapped image is formed according to the following rule:

$$M_{k,l} = \begin{cases} 1, & D_{k,l}(i,j) < -\beta \\ 0, & -\beta \leq D_{k,l}(i,j) \leq \beta \\ 1, & D_{k,l}(i,j) > \beta \end{cases} \quad (3.1)$$

where $1 \leq i \leq 3$, and $1 \leq j \leq 3$, $(i,j) \neq (2,2)$.
The threshold parameter β is adaptive and is given by,

$$\beta = \left[41 - 0.00234(D_{k,l}(i,j) - 127.5)^2\right] \quad (3.2)$$

In case of salt-and-pepper noise, maximum and minimum pixel values are 255 and 0 respectively. If a center pixel has maximum or minimum value, then β value reaches to its minimum value.

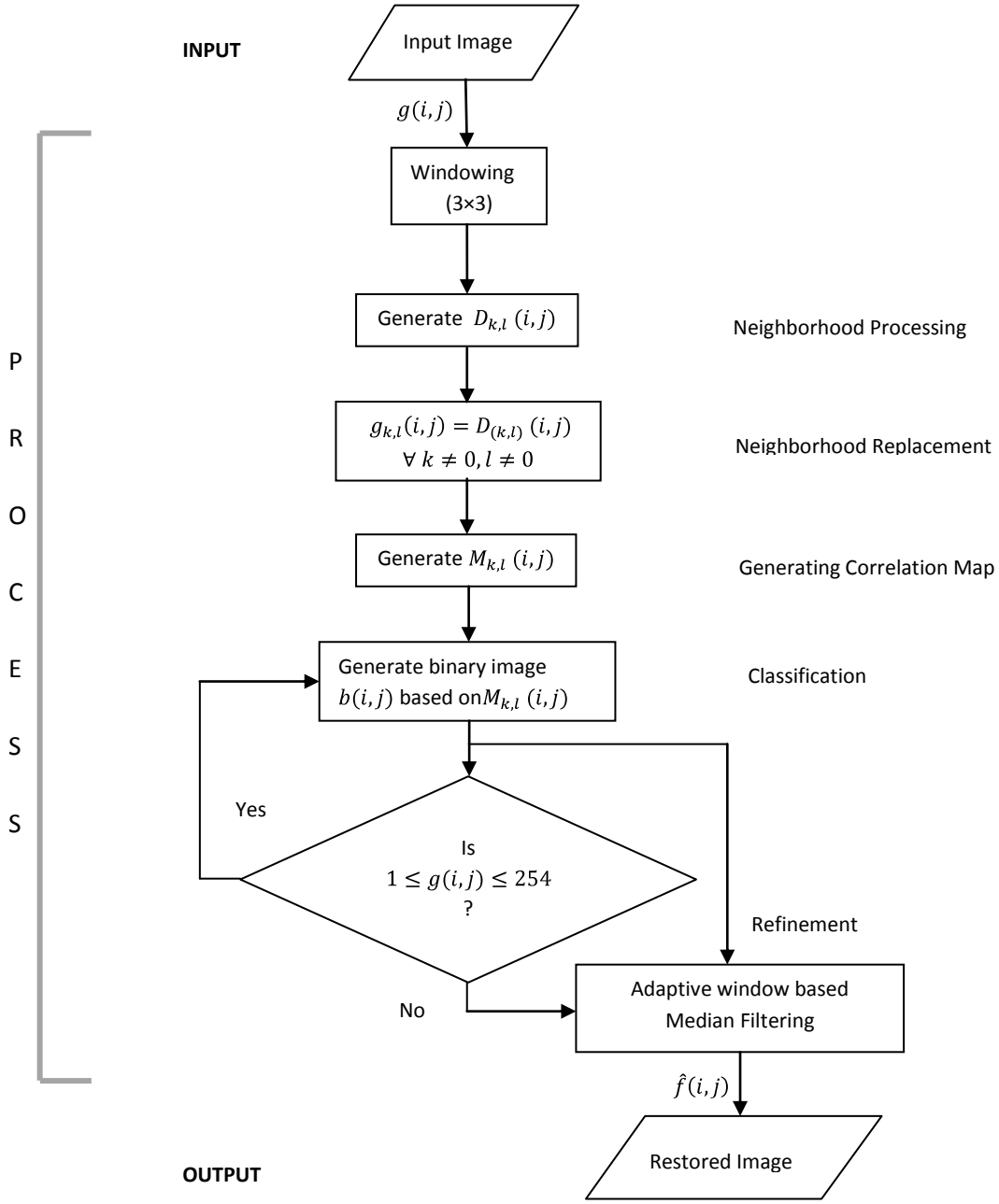


Fig. 3.2 Flow chart for noise detection

Step 3: Classification of Pixel

Initially all pixels of $g(i,j)$ are labeled as noise-free pixels in a binary flag image $b(i,j)$ of size M×N, that is all values are set to zero initially. From the correlation map $M_{k,l}$ central pixel will be classified as noisy or noise-free, based on the number of zeros (Z) in the eight neighborhood of $M_{k,l}(i,j)$. If $Z \geq 3$ then current pixel $g(i,j)$ is classified as a noise free and $b(i,j) = 0$ otherwise $b(i,j) = 1$. On the other hand, Z will be small when the noise density is high.

Step 4: Refinement

Elements of $b(i,j)$ give information whether a pixel has been classified as noisy or noise-free. Since salt-and-pepper corrupted pixels have values 0 and 255 respectively, then the binary flag matrix $b(i,j)$ will be subjected to further verification as per the proposition given below.

Proposition: If a pixel $g(i,j)$ satisfies the condition

$$1 \leq g(i,j) \leq 254 \tag{3.3}$$

Then the pixel is declared as *noise-free* and thus $b(i,j)$ will be retained as ***0*** if it is assigned a value ***1*** in the previous stage.

3.2.2 Adaptive Noise Filtering

Fig. 3.3 shows the flowchart for adaptive noise filtering. Based on the binary flag, *no filtering* is applied to those *uncorrupted* pixels (i.e. $b(i,j) = 0$ while the SM (switching median) with an adaptively determined window size is applied to each *corrupted* one (i.e. $b(i,j) = 1$).

The *maximum window size* is limited to (7×7) in order to avoid severe blurring of image details at high noise density cases. Starting with (3×3) filtering window iteratively extends outward by one pixel in all the four sides of the window, provided that the number of uncorrupted pixels is less than *half* of the total number of pixels within the filtering window. Only the pixels that are classified as noise free in filtering window will participate in median filtering process. This will, in turn, yield a better filtering result with less distortion.

Intensive simulations are carried out using several monochrome test images, which are corrupted with impulse noise of various noise densities. The simulation results are presented in Section-3.8.

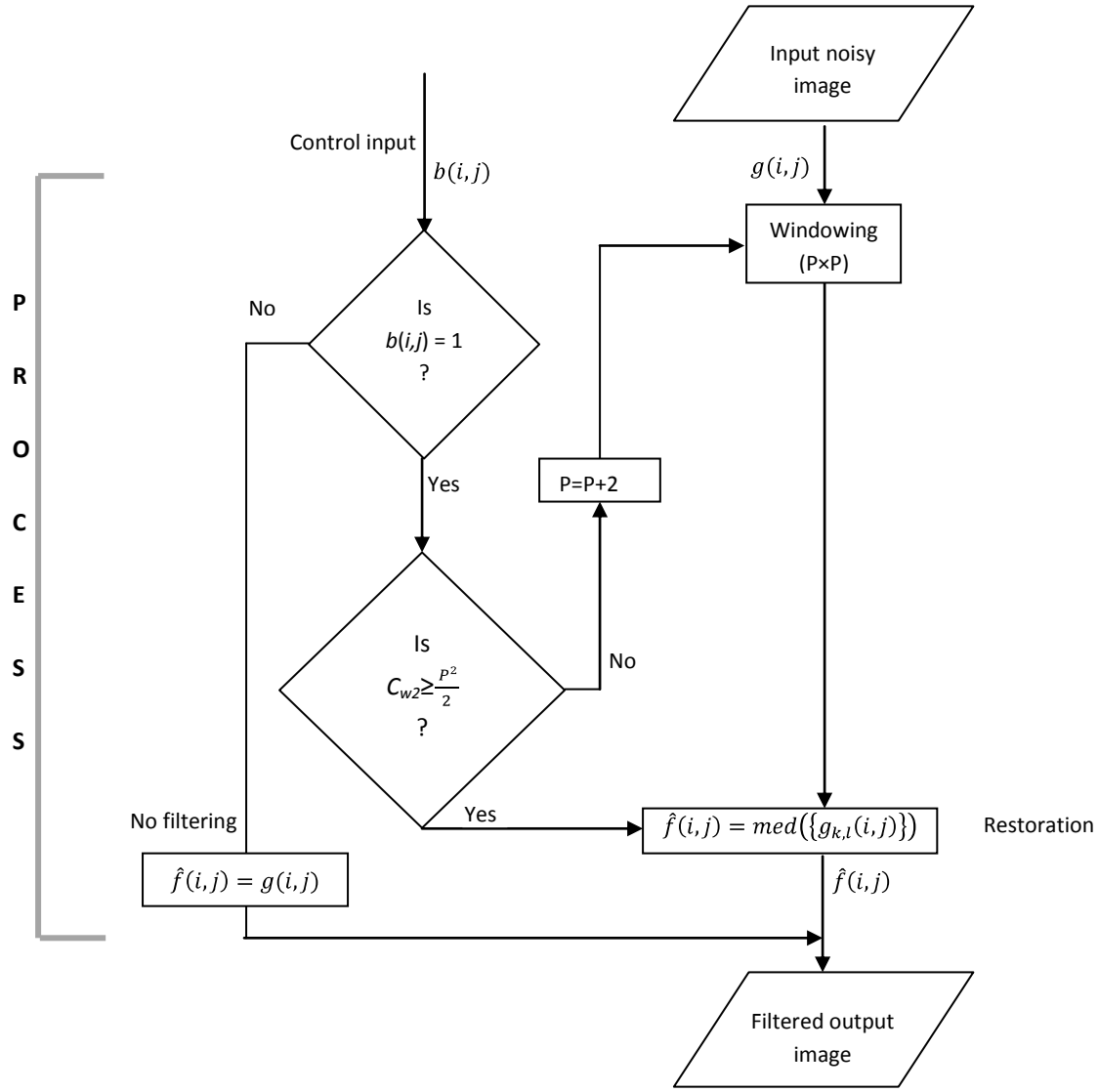


Fig. 3.3 Flow chart for adaptive filtering

3.3. Robust Estimator based Impulse-Noise Reduction (REIR) Algorithm [P2]

A robust statistical estimator, Lorentzian estimator [78], is employed in ICF-2 paradigm as shown in Fig. 3.1(d) in Section3.1, where adaptive window is used for pixel classification. The noisy pixel is replaced with Lorentzian estimator or average of the previously processed pixels. Because of adaptive windowing technique, the filter is able to suppress the noise at a density as high as 90%.

3.3.1 Background

In recent times, nonlinear estimation techniques have been gaining popularity in image denoising problems. But they fail to remove noise in high frequency regions such as edges and fine details in the image.

To overcome this problem a nonlinear estimation technique has been developed based on robust statistics. The contaminating noise in an image is considered as a violation of assumption of spatial coherence of the image intensities and is treated as an outlier random variable [51, 77]. When the ideal assumptions of a system are violated, problem of estimation can be solved by robust statistics techniques. A robust estimation based filter [51] is available in literature that suppresses the low-to-medium density additive noise quite efficiently. Being encouraged with its performance, the same basic concept of robust estimation filter [51] is modified and implemented in an adaptive windowing framework to suit the fixed-valued impulse noise suppression application.

Robustness is measured using two parameters: *influence curves* and *breakdown point*. The influence curves tell us how an infinitesimal proportion of contamination affects the estimate in large samples. The breakdown point is the largest possible fraction of observations for which there is a bound on the change of the estimate when that fraction of the sample is changed without restrictions.

If an estimator is more forgiving about outlying measurements, then robustness increases. In the proposed method, a re-descending estimator is considered for which the influence of outliers tends to zero with increasing distance. A Lorentzian estimator [78] has an influence function which tends to zero for increasing estimation distance and maximum breakdown value. Therefore, it is employed to estimate an original image pixel from noise corrupted pixel in the proposed filer.

The Lorentzian estimator and its influence function are given by:

$$\rho(x) = log\left(1 + \frac{x^2}{2\sigma^2}\right) \tag{3.4}$$

$$\varphi_{lorentz}(x) = \frac{2x}{(2\sigma^2 + x^2)} \tag{3.5}$$

where x is Lorentzian estimation distance and σ is standard deviation.

An image is assumed to be non-stationary. Hence, the image pixels are sampled with small spatial windows (3×3, 5×5 or 7×7) and this estimation algorithm is applied to each window.

3.3.2 Proposed Algorithm

Fig. 3.4 shows the flowchart of the proposed algorithm. Let $g(i,j)$ denotes a corrupted image. For each pixel (i,j), a 2-D sliding window $g_{k,l}(i,j)$ is selected such that the current pixel (i,j) lies at the center of the sliding window. Let g_{min}, $m(i,j)$ and g_{max} be the minimum, median and maximum gray level in the selected window. Let (P×Q) be the window size. In this case a square window is used where P = Q.

The proposed algorithm is as follows:

Step-1. Initialize the sliding window size, P to 3.

Step-2. Determine g_{min}, $m(i.j)$ and g_{max} in $g_{k,l}(i,j)$.

Step-3. **IF** $g_{min} < m(i,j) < g_{max}$,

GO TO Step-5

ELSE increment window size, P to P+2, provided P ≤ 7.

Step-4. **IF** P ≤ 7,

GO TO step 2,

ELSE replace the center pixel with the mean of the processed neighborhood pixel values.

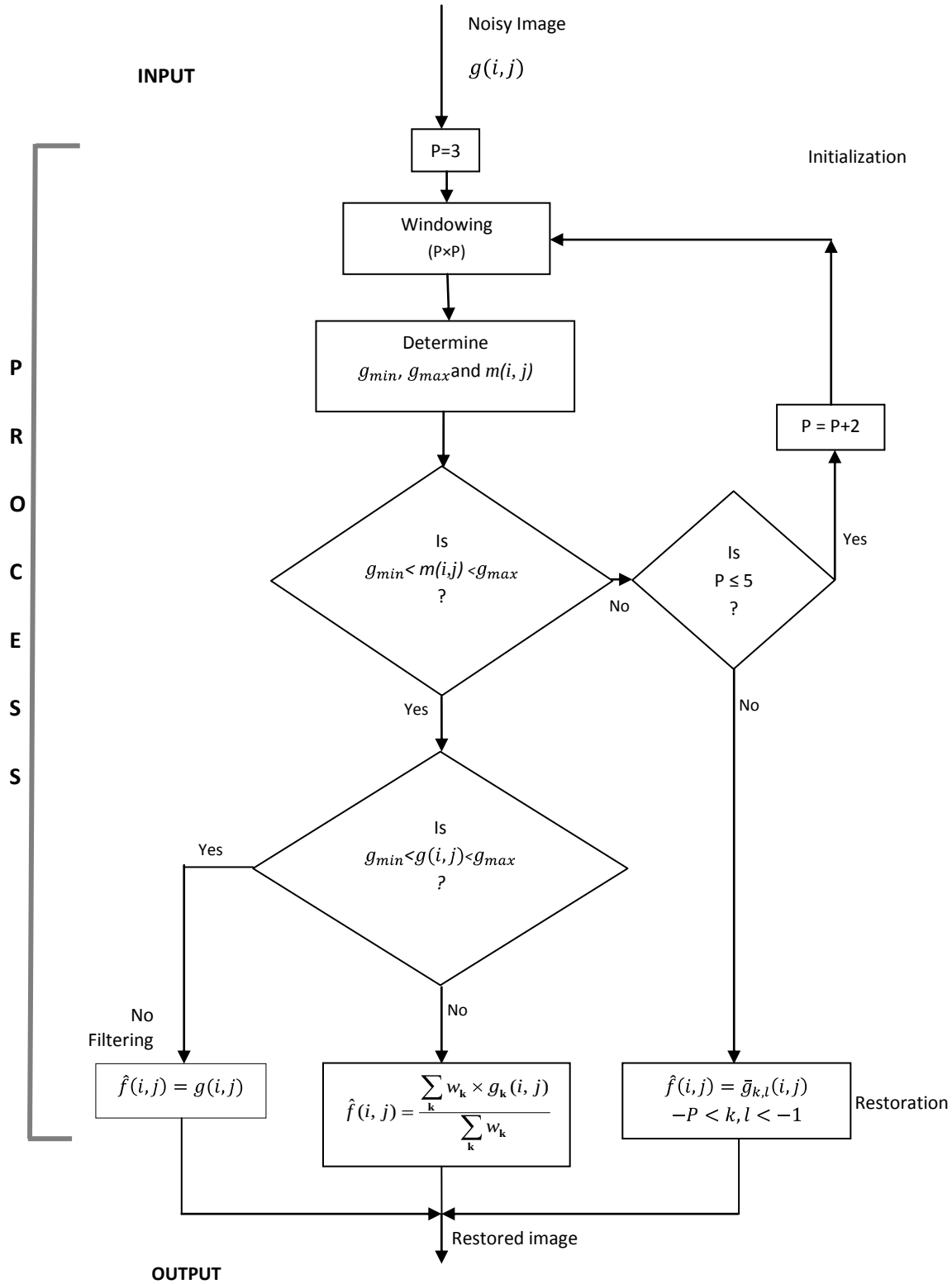


Fig. 3.4 Flowchart for REIR

Step-5. **IF** $g_{min}<g(i,j)<g_{max}$,

THEN $g(i,j)$ is considered a *noise-free* pixel,

ELSE select a pixel in the window such that $g_{min}<g_{k,l}(i,j)<g_{max}$

Step-6. Compute **A**bsolute **D**eviation from **M**edian (ADM), $d_{\boldsymbol{k}}$ **,** $\boldsymbol{k}$ = (k,l), defined by

$$d_{\boldsymbol{k}} = \left|g_{k,l}(i,j) - m\right| . \tag{3.6}$$

Step-7. Compute influence function ψ(.) as follows.

$$\psi(d_{\boldsymbol{k}}) = \frac{2d_{\boldsymbol{k}}}{\left(2\sigma^2 + d_{\boldsymbol{k}}{}^2\right)} \tag{3.7}$$

where σ is outlier rejection point, given by,

$$\sigma = \frac{\tau_s}{\sqrt{2}} \tag{3.8}$$

where τ_s is maximum expected outlier, which is calculated as,

$$\tau_s = \zeta\sigma_N \tag{3.9}$$

where σ_N is the local estimation of the image standard deviation, where $\boldsymbol{\zeta}$ is a smoothening factor and is chosen as 0.3 for low to medium smoothing.

Step-8.filtered image is estimated by

$$\hat{f}(i,j) = \frac{\sum_{\mathbf{k}} w_{\mathbf{k}} \times g_{\mathbf{k}}(i,j)}{\sum_{\mathbf{k}} w_{\mathbf{k}}} \tag{3.10}$$

where $w_{\mathbf{k}} = \dfrac{\Psi(d_{\mathbf{k}})}{d_{\mathbf{k}}}$

An exhaustive simulation work is carried out and results are presented in Section-3.8.

3.4. Impulse Denoising Using Improved Progressive Switching Median Filter (IDPSM) [P3]

The proposed algorithm is developed based on ICF-1 paradigm as shown in Fig. 3.1(c) in Section 3.1. The noisy pixel is replaced with median of uncorrupted pixels in an adaptive filtering window. The iterative nature of the filter makes it more efficient in noise detection and adaptive filtering window technique makes it robust enough to preserve edges and fine details of an image in wide range of noise densities.

3.4.1 Impulse Noise Detection

Fig 3.5 shows the flowchart for noise detection algorithm. The flowchart itself explains the complete noise detection processes. After n-iterations the algorithm generates binary flag image $b^{(n)}(i,j)$. Where n is positive integer (n$\in \mathcal{Z}^{+}$).

Let $g(i,j)$ be the input noisy image. For each pixel (i,j), a 2-D sliding window $g_{k,l}(i,j)$ of size 3×3 is selected such that the central pixel (i,j) lies at the center of the sliding window. The algorithm is explained as follows:

Let n be the number of iteration. Initialize the iteration index I=1, and binary flag image $b(i,j) = 0$. Calculate g_{min}, g_{max} and m for selected window $g_{k,l}(i,j)$.

IF $(\left|g(i,j) - m\right| < T \; AND \; g_{min} < g(i,j) < g_{max}$

THEN $b(i,j) = 0$

ELSE $b(i,j) = 1$.

This process is repeated for complete image, and the complete algorithm is repeated until iteration index I = n (i.e., up to n-iterations).

3.4.2 Refinement

Elements of $b^{n}(i,j)$ give the information whether the pixel has been classified as noisy or noise-free. Since salt-and-pepper has minimum and maximum pixel values 0 and 255 respectively, the binary flag image is cross-checked. If any pixel has been classified as noisy but its value will be in the range (0,255), then the corresponding flag is changed from 1 to 0. This improves the performance of filtering algorithm.

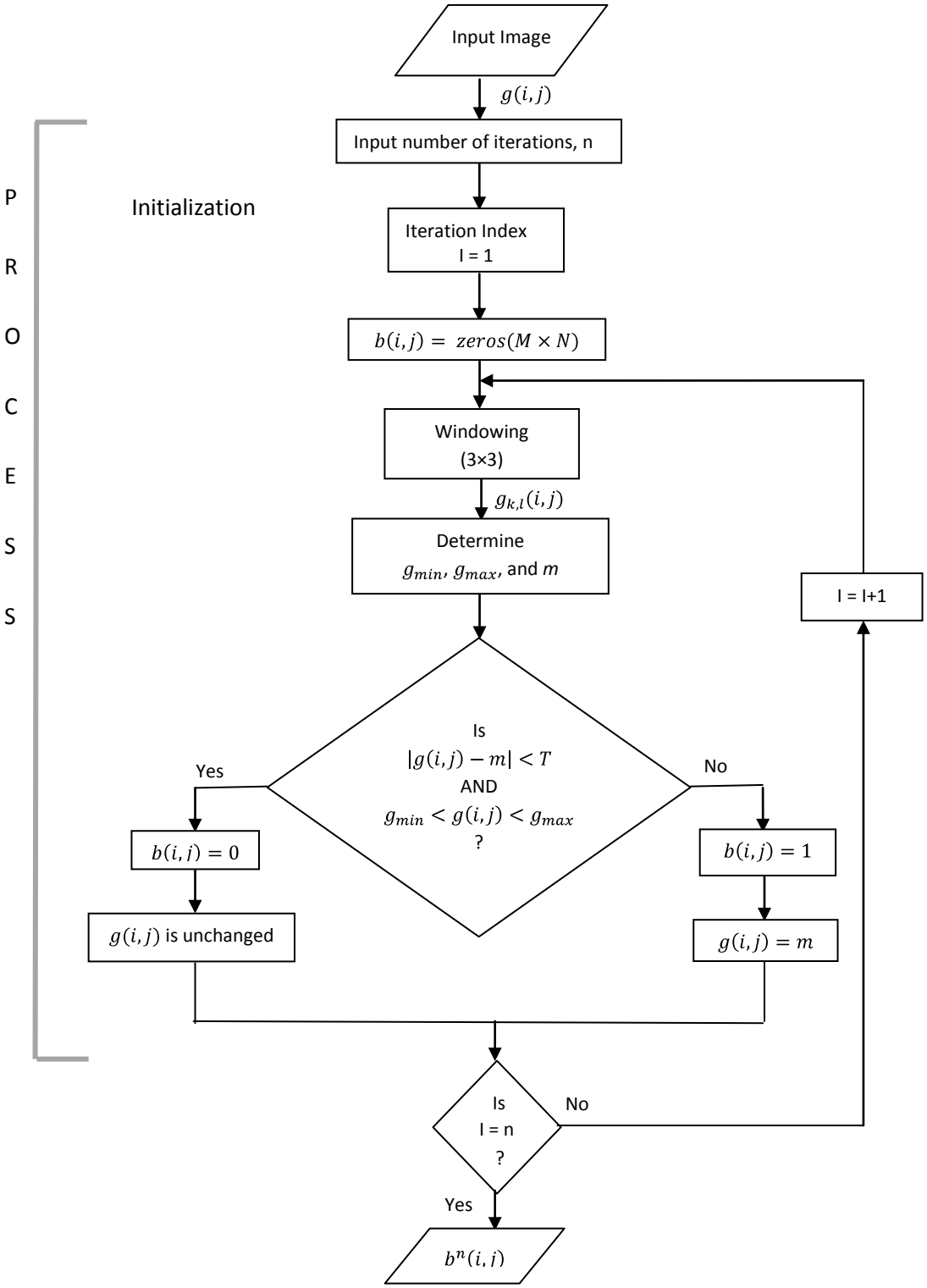


Fig. 3.5 Flowchart for noise detection

3.4.3 Noise Filtering

Fig 3.6 explains the filtering algorithm. The filtering algorithm takes two 2-D signals as its input. In addition to the noisy image $g(i,j)$, it also accepts the binary flag image $b(i,j)$. In fact, binary flag image $b(i,j)$ controls the filtering process and, therefore, may be considered as a control signal. Let us introduce the filtering window $g_{k,l}^{(x)}(i,j)$ with size $P^x \times P^x$, and $P^x = 3 + 2x$, where x = 0,1 and 2. Initialize x = 0 and calculate $q^{(x)}$. Let $m^x(i,j)$ be the median of noise-free pixels and Cw_2 be the number of noise-free pixels in the filtering window. If $Cw_2 \geq q^{(x)}$, then replace the noisy pixel with $m^x(i,j)$ in the filtered image$\hat{f}(i,j)$. Otherwise increment the x by 1. If $x > 2$, then replace the noisy pixel with left neighboring pixel of center pixel in the output image $\hat{f}(i,j)$. Otherwise recalculate $q^{(x)}$, $m^x(i,j)$ and Cw_2 for new filtering window and repeat the above process.

The value of threshold T is important whose optimum value is evaluated searching for best performance in terms of PSNR in separate experiment, discussed in Section-3.4.4. The noise ratio γ is given by:

$$\gamma = \frac{C_1}{C_1 + C_2} \tag{3.11}$$

where $C_1$is total number of noisy pixels and C_2 is total number of noise-free pixels in the image. Thus, the total number of pixels is represented by (C_1+C_2). The value of γ lies between 0 and 1 (i.e.,$0 \leq \gamma \leq 1$). The parameter, $q^{(x)}$ for x = 0, 1, 2 are defined as:

$$q^{(0)} = a_0 + b_0 \times \gamma \tag{3.12}$$

$$q^{(1)} = a_1 + b_1 \times \gamma \tag{3.13}$$

$$q^{(2)} = a_2 + b_2 \times \gamma \tag{3.14}$$

The other parameters are given by, $a_0 = 4, b_0 = 3, a_1 = 13, b_1 = 5, a_2 = 26, b_2 = 6$.

3.4.4 Optimizing the Threshold

In order to optimize the value of threshold, a number of simulation experiments are conducted on standard test images, corrupted with SPN of different noise densities. The performance is evaluated in terms of PSNR. The simulated results of *Lena* test image is tabulated in table Table-3.1. It is observed that the proposed system yields high performance, in terms of PSNR, for the threshold, T ∈ [40, 50]. Thus, an optimized value of threshold, T, i.e., $T_{optimal}$ taken is 45.

Extensive simulations are conducted on the different gray scale test images and simulation results are presented in Section 3.8.

Table-3.2: Performance of IDPSM filter in terms of PSNR for different Threshold, T operated on Lena image corrupted with SPN under various noise densities

Sr.No	Threshold T	SPN Noise (%)		
		20	**30**	**40**
1	10	30.63	29.03	28.07
2	20	33.85	31.41	29.56
3	30	37.02	33.31	30.41
4	40	38.19	34.96	30.97
5	**45**	**38.24**	**35.12**	**31.62**
6	50	38.22	35.10	31.59
7	60	37.02	34.74	30.79
8	70	35.63	32.77	30.01

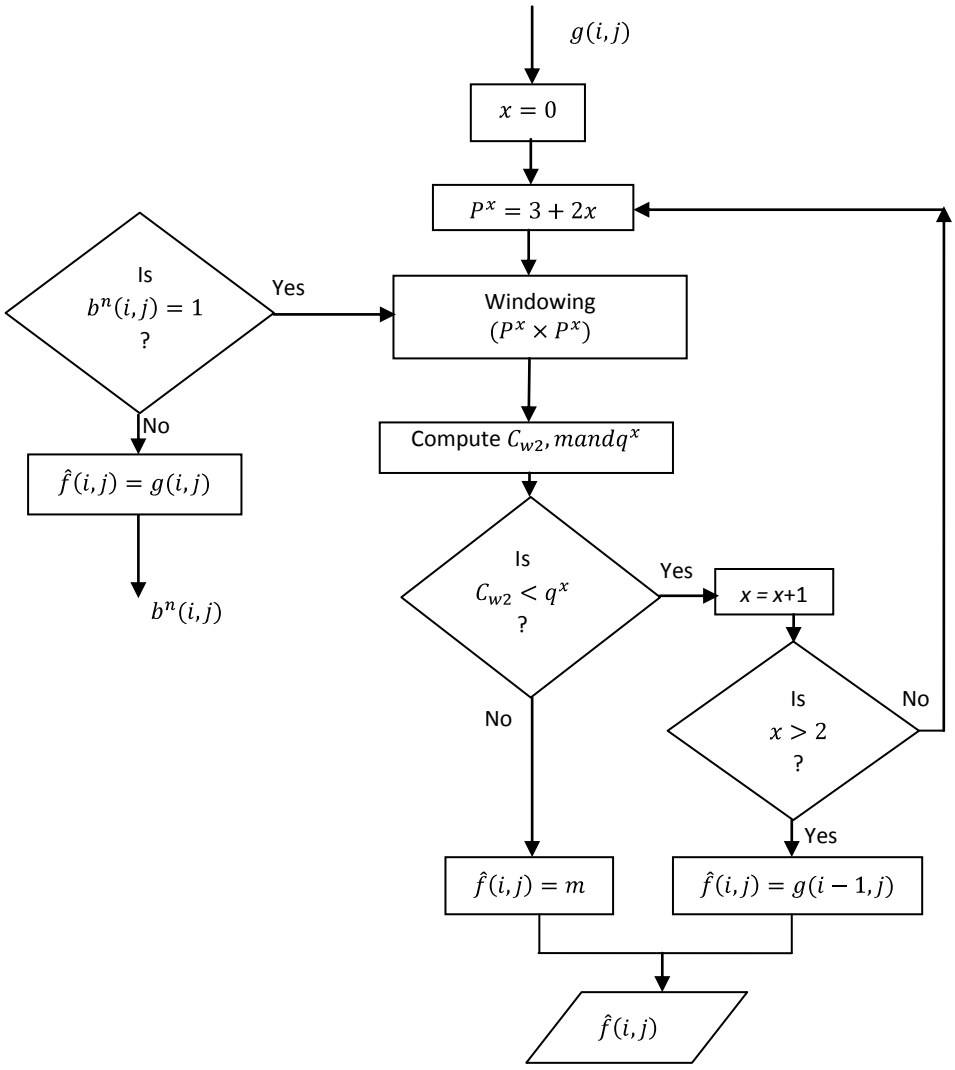


Fig. 3.6 Flowchart for filtering

3.5. Impulse-Noise Removal by Impulse Classification (IRIC) [P4]

This algorithm is developed under the framework of SF paradigm as shown in Fig. 3.1(a) in Section-3.1. The noisy pixel is replaced with median of processed pixels in the filtering window. At high noise densities, the median filtering may not be able to reject outliers always. Under such circumstances, the processed left neighboring pixel is considered as the estimated output. The computational complexity of this method is equivalent to that of a median filter having a 3×3 window. The proposed algorithm requires simple physical realization structures. Therefore, this algorithm may be quite useful for online and real-time applications.

3.5.1 Proposed Algorithm

The algorithm is explained below.

Step-1. Select a window $g_{k,l}(i,j)$, $-1 \leq k,l \leq 1$.

Step-2. **IF** $0 < g(i,j) < 255$

GO TO Step-3

ELSE GO TO Step-4.

Step-3. **No Filtering:**

$\hat{f}(i,j) = g(i,j)$

EXIT.

Step-4. **Estimation:**

Determine median, $m = med\left\{g_{k,l}(i,j)\right\}$.

Step-5. **Filtering Process:**

IF $0<m<255$

$\hat{f}(i,j) = m$

ELSE $\hat{f}(i,j) = g(i-1,j)$.

Step-6. Repeat Step-1 to Step-5 for all (i,j)locations,

END.

The flowchart of the proposed algorithm is depicted in Fig. 3.7.

The performance of the algorithm is tested with different gray scale images. The simulated results are presented in Section-3.8.

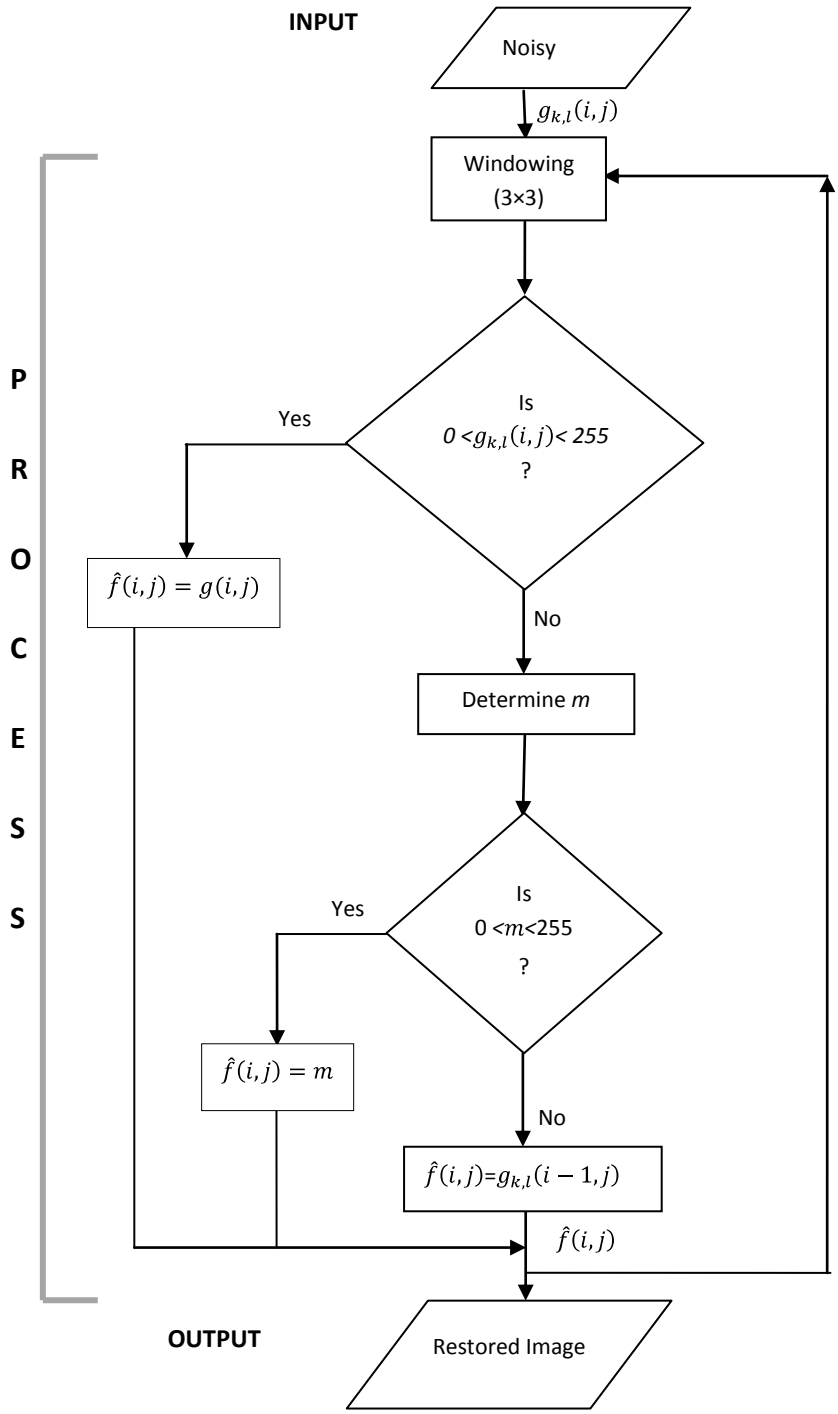


Fig. 3.7 Flowchart for IRIC algorithm

3.6. Adaptive Switching Filter (ASF) [P5-P6]

Two different adaptive switching filters: **ASF-I** and **ASF-II** are developed by using BCF paradigm, as shown in Fig. 3.1(b) in Section-3.1, for suppressing high density SPN. The noisy pixel is replaced with *alpha-trimmed mean* value of uncorrupted pixels in the adaptive filtering window. Depending on noise estimation, a small filtering window size is initially selected and then the scheme adaptively changes the window size based on the number of *noise-free* pixels. Therefore, the proposed method removes the noise much more effectively even at noise density as high as 90% and yields high image quality.

3.6.1 A Novel Adaptive Switching Filter (ASF-I) [P5] for Suppression of High Density SPN

The proposed method uses fixed window size, 3×3, for noise detection and an adaptive window for filtering.

Noise Classification:

Fig. 3.8 shows the flowchart for noise detection which is self explanatory. The noise detection algorithm can be glanced as follows:

IF $g_{min} < g(i,j) < g_{max}$

THEN $b(i,j) = 0$

ELSE $(g(i,j) = g_{min})\ AND\ (g_{min} > 0)$

THEN $b(i,j) = 0$

ELSE $(g(i,j) = g_{max})\ AND\ (g_{max} < 255)$

THEN $b(i,j) = 0$

ELSE $b(i,j) = 1$

END.

Filtering:

The estimation (filtering) process adopted in this algorithm is given by (2.6).

Here, $m(i,j)$ is the alpha-trimmed mean value obtained from adaptive filtering window.

Fig. 3.9 represents the flowchart for image restoration process. Square filtering window (i.e. P = Q) with odd dimension employed here is given by,

$$P = Q = 2r + 1, \text{ where, } r \in \mathcal{Z}^{+}. \tag{3.15}$$

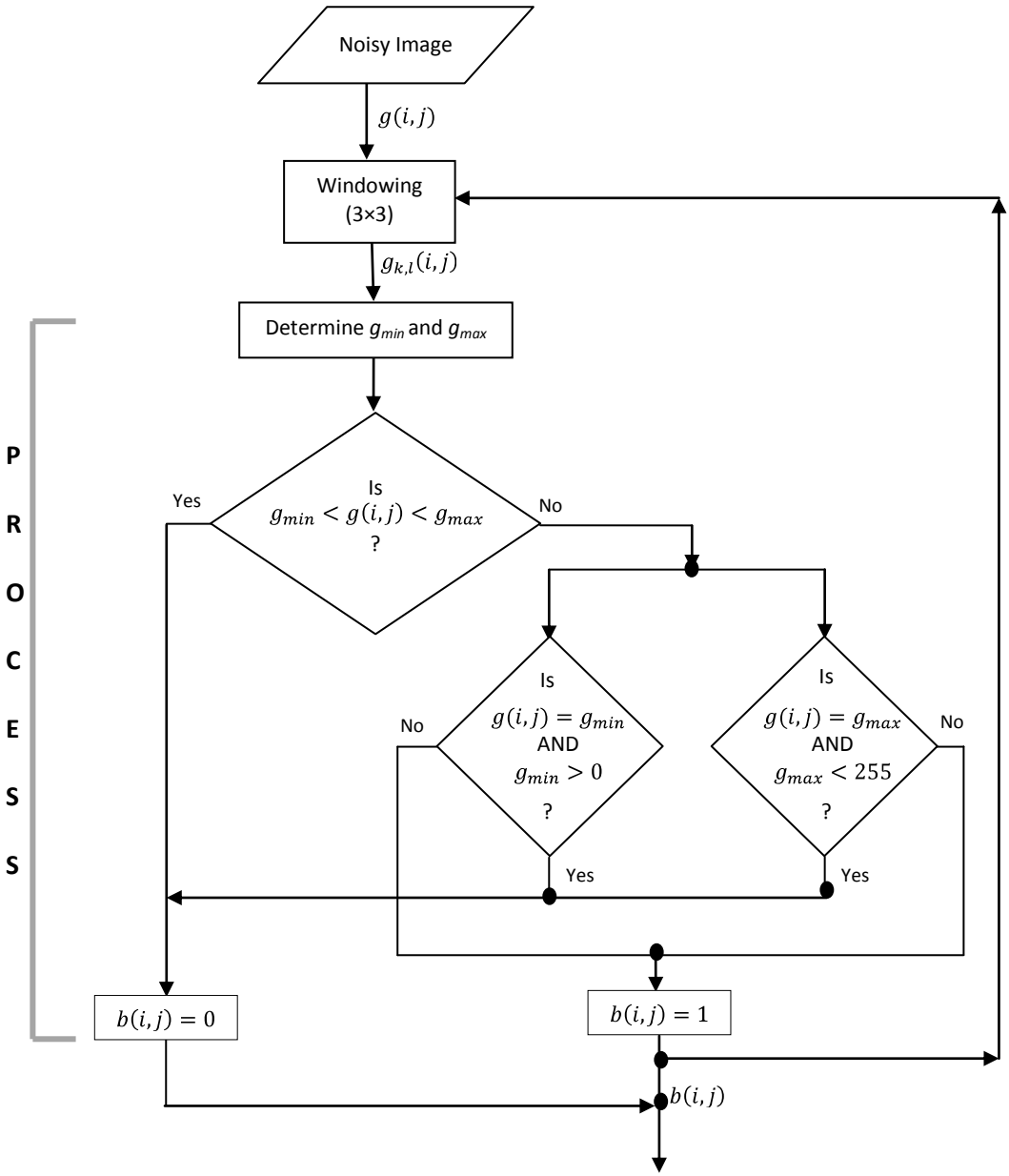


Fig. 3.8 Flowchart for noise detection

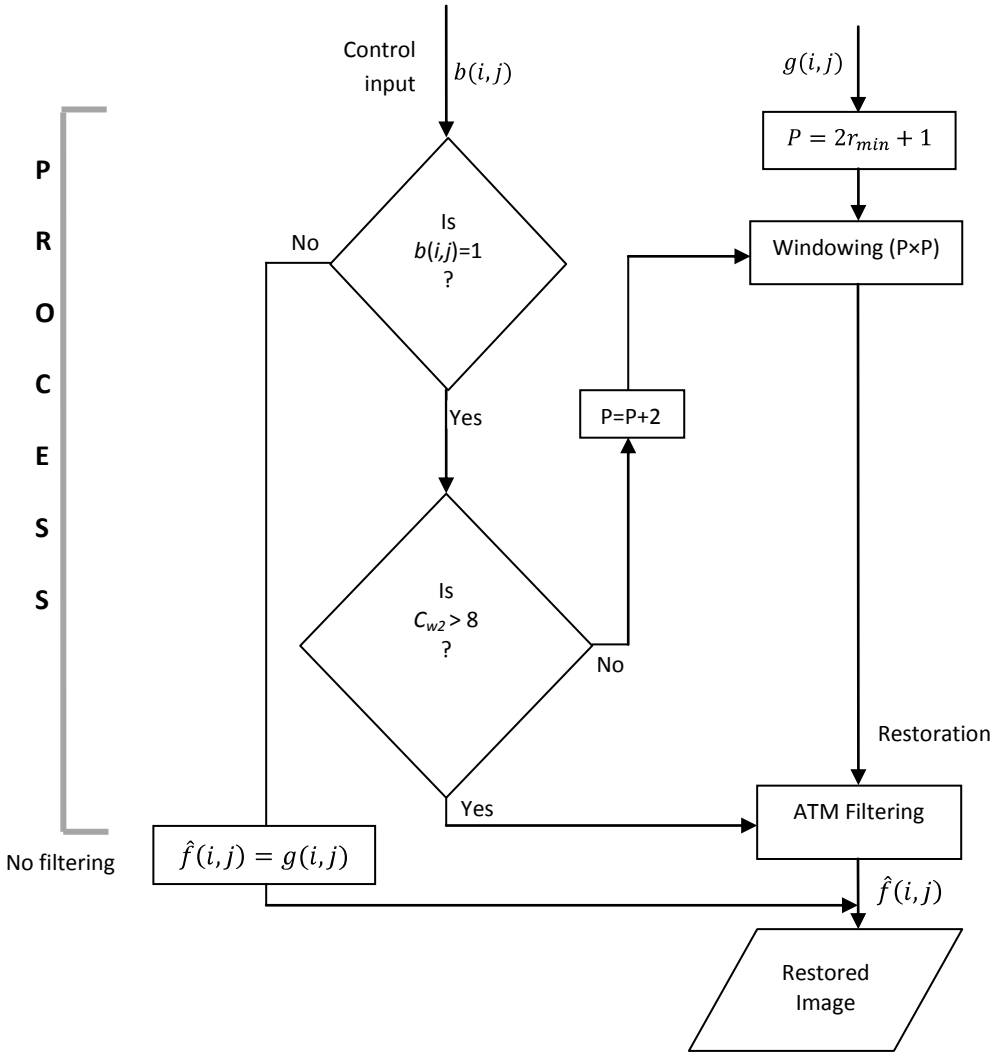


Fig. 3.9 Flowchart for restoration

To determine the value of $m(i,j)$ a rule is set that the minimum number of *noise-free* pixels needed for this calculation must be greater or equal to *eight* pixels. If a small sample size is taken, where the total *noise-free* pixels are less than eight, these samples are not good enough to present the local information of the image properly when the noise level is high. If the size is too big that is also not enough to present the local information of the image properly.

The r_{min} (minimum size of filtering window) is calculated as follows.

Let C_1 be the number of noisy pixels that have been detected (i.e., number of **1**s present in the binary flag image $b(i,j)$) and C_2 be the number of noise-free pixels in the image. Thus, the total number of pixels is represented by C_1+C_2. The impulse noise density γ is estimated as,

$$\gamma = \frac{C_1}{C_1 + C_2} \tag{3.16}$$

The value of γ lies between 0 and 1 (i.e., $0 \leq \gamma \leq 1$).

In order to minimize the number of trials needed to find the correct filter size, the value of $r_{min}$is calculated by using following equation,

$$r_{min} = \left\lfloor \sqrt{\frac{2}{1-\gamma}} \right\rfloor \tag{3.17}$$

The symbol $\lfloor . \rfloor$ represents *floor* operation. By using r_{min}, the algorithm converges faster, because less iterations are needed to find the correct window size for filtering.

The novel adaptive method for finding $m(i,j)$is described as follows.

Step-1. Initialize the size of filtering window, $P = 2r_{min} + 1$, where r_{min} is a small integer value, defined by (3.17).

Step-2. Compute the *noise-free* pixels C_{w2} in the filtering window $g_{k,i}(i,j)$ of size P x P.

Step-3. **IF** $C_{w2} < 8$,

P = P+2

GO TO Step-2

ELSE, GO TO Step-4

Step-4. Compute $m(i,j) = ATM(\{g_{k,i}(i,j)\})$

where ATM (.) is the alpha-trimmed mean (2.1)

Step-5 Update the value of $\hat{f}(i,j)$by using (3.15).

The simulated results are presented in Section 3.8.

3.6.2 A Novel Adaptive Switching Filter (ASF-II) [P6] for Suppression of High Density SPN

This is a modified version of the adaptive switching filter, ASF-I, presented in Section-3.5.1. A different filtering process is adopted here so that the algorithm suits well for high-density SPN. To determine the value of $m(i,j)$ a condition is set such that, the minimum number of *noise-free* pixels, C_{w2}, needed for this calculation must be *greater than or equal to half of the total number of pixels in the filtering window*. The small number of samples is not good enough to present the local information of the image when the noise density is high. If the number of samples are increased, the

size of filtering window increases, which introduces blurring effect in an image. On the other hand, if less samples are taken, the size of filtering window reduces, which may not filter the noise properly. Fig. 3.10 shows the flowchart for the filtering process. The simulation results are presented in Section-3.8.

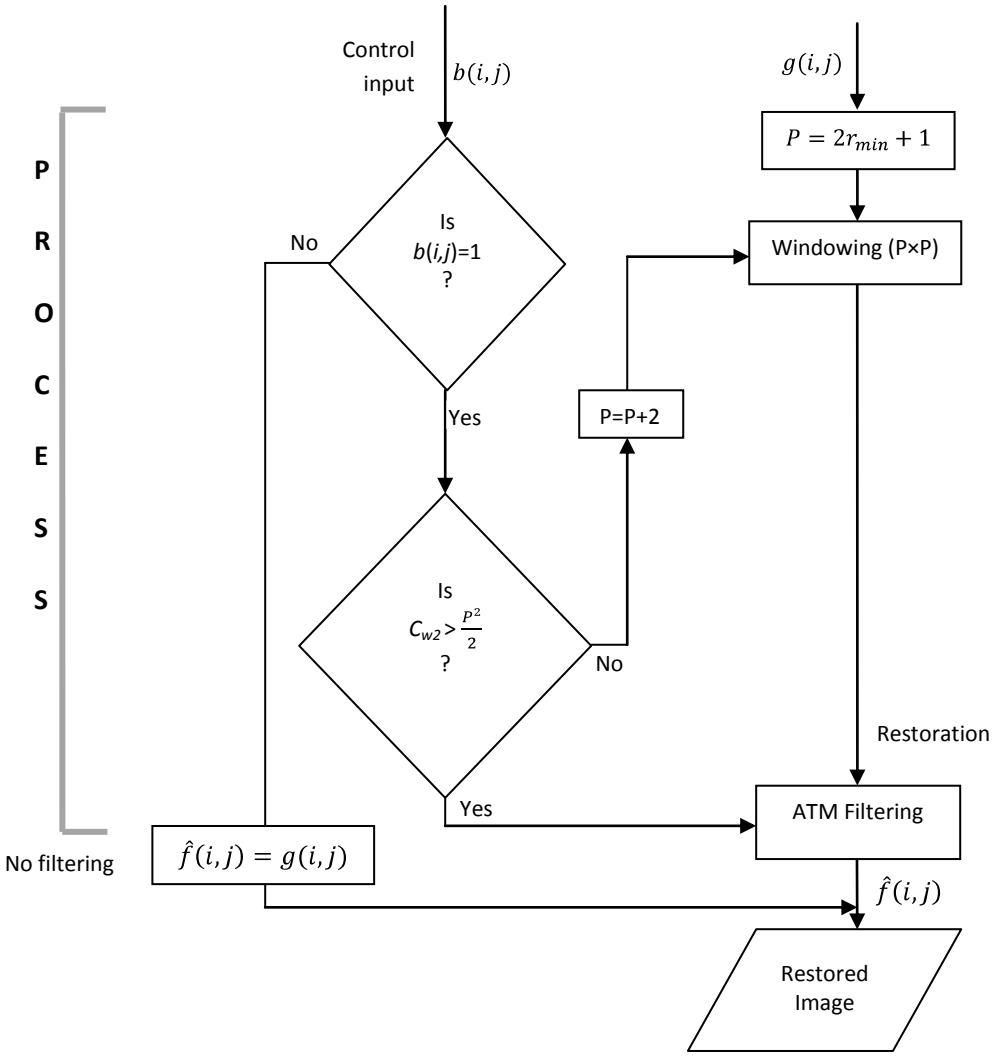


Fig. 3.10 Flowchart for restoration

3.7. Impulse Denoising Using Iterative Adaptive Switching Filter (IASF) [P7]

The developed algorithm employs the framework of ICF-1 as shown in Fig. 3.1(c) in Section-3.1. The noisy pixel is replaced with *alpha-trim mean* value of uncorrupted pixels in the adaptive filtering window. Due to its iterative structure, the performance of this filter is better than existing order-statistic filters. Further, the adaptive filtering window makes it robust enough to preserve the edges and fine details of an image.

3.7.1 Detection of noisy pixels

Let $g(i,j)$ and $b(i,j)$be the noisy input image and binary flag image respectively. Let them be of size (M×N).The noisy pixel is represented by binary flag $b(i,j)$=1 and noise-free pixel is represented by $b(i,j)$= 0. A variable C_2, which is non-negative integer *initialized to zero*, is used to determine the number of valid non-impulsive pixels in the current iteration. Initially, all the pixels are assumed to be impulse, so the binary flag image $b(i,j)$is set to *unity*.

The flowchart for proposed algorithm is shown in Fig. 3.11 and it is explained below:

Step-1. Initialize: $b(i,j)$= ones (M×N), C_2 = 0, and variable flag = 0.

Step-2.Select a window $g_{k,l}(i,j)$,$-1 \leq k,l \leq 1$.

Step-3. Determine g_{min} (minimum) and g_{max} (maximum) in the selected window.

Step-4. Compute the parameters:

A1 = $g(i,j) - g_{min}$; A2 =$g(i,j) - g_{max}$

Step-5. **IF** A1 > 0 and A2 < 0

THEN $b(i,j)$= 0

$C_2 = C_2 + 1$

ELSE $b(i,j)$ unchanged.

Step-6. **IF** flag = 0,

THEN, *flag* =1; $g(i,j) = g_{min}$

ELSE

flag = 0; $g(i,j) = g_{max}$

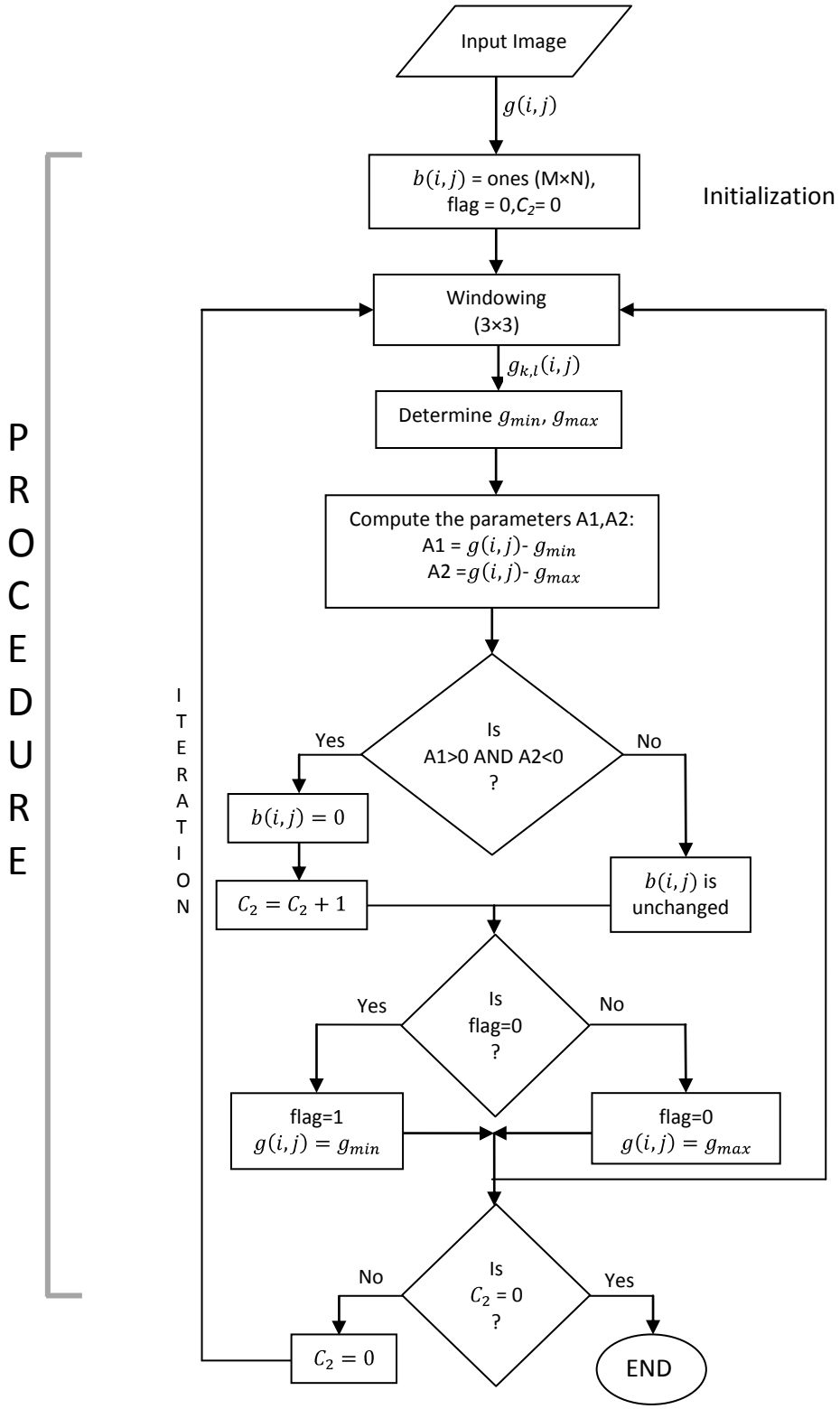


Fig. 3.11 Flowchart for noise detection

Step-6. Check the count C_2:

IF count $C_2 \neq 0$,

THEN, reset count $C_2 = 0$; **GO TO** Step-2,

ELSE

END the iteration.

If the centre pixel $g(i,j)$ lies between minimum and maximum values of a window, then the current pixel is non-impulsive which can be retained at the output image, and the flag at that position is reset to 0 and C_2 is incremented by one. Here the pixel $g(i,j)$ is replaced with minimum or maximum value of the current window according to a *flag* value which switches to 0 or 1 alternatively so as to propagate the impulse or impulse like noise throughout the entire image. This replacement of $g(i,j)$with minimum or maximum value will prevent the other pixels surrounding $g(i,j)$being wrongly identified as an impulse. This process is continued for all the pixels in the image. The value of variable C_2 at the end of current iteration will give the number of pixels newly detected as valid pixels which can be used for checking whether to stop or continue the iteration process. The algorithm will continue the iteration process until the value of the variable C_2 converges to zero.

3.7.2 Impulse noise correction

The output filtered image is estimated by following relation:

$$\hat{f}(i,j) = \begin{cases} g(i,j), & b(i,j) = 0 \\ m(i,j), & b(i,j) = 1 \end{cases} \tag{3.18}$$

An adaptive window is used for the estimation of $m(i,j)$. Fig 3.12 shows the flowchart for the noise suppression which is self explanatory. The size of filtering window is incremented by two, based on the local information of the filtering window. The minimum filtering size is determined by,

$$P = Q = 2r + 1 \tag{3.19}$$

where, $r \in \mathcal{Z}^+$, r_{min} value is given by (3.17)

The procedure for calculation of r_{min} is explained completely in the proposed algorithm, Adaptive Switching Filter-I (ASF-I), in Section-3.5.1.

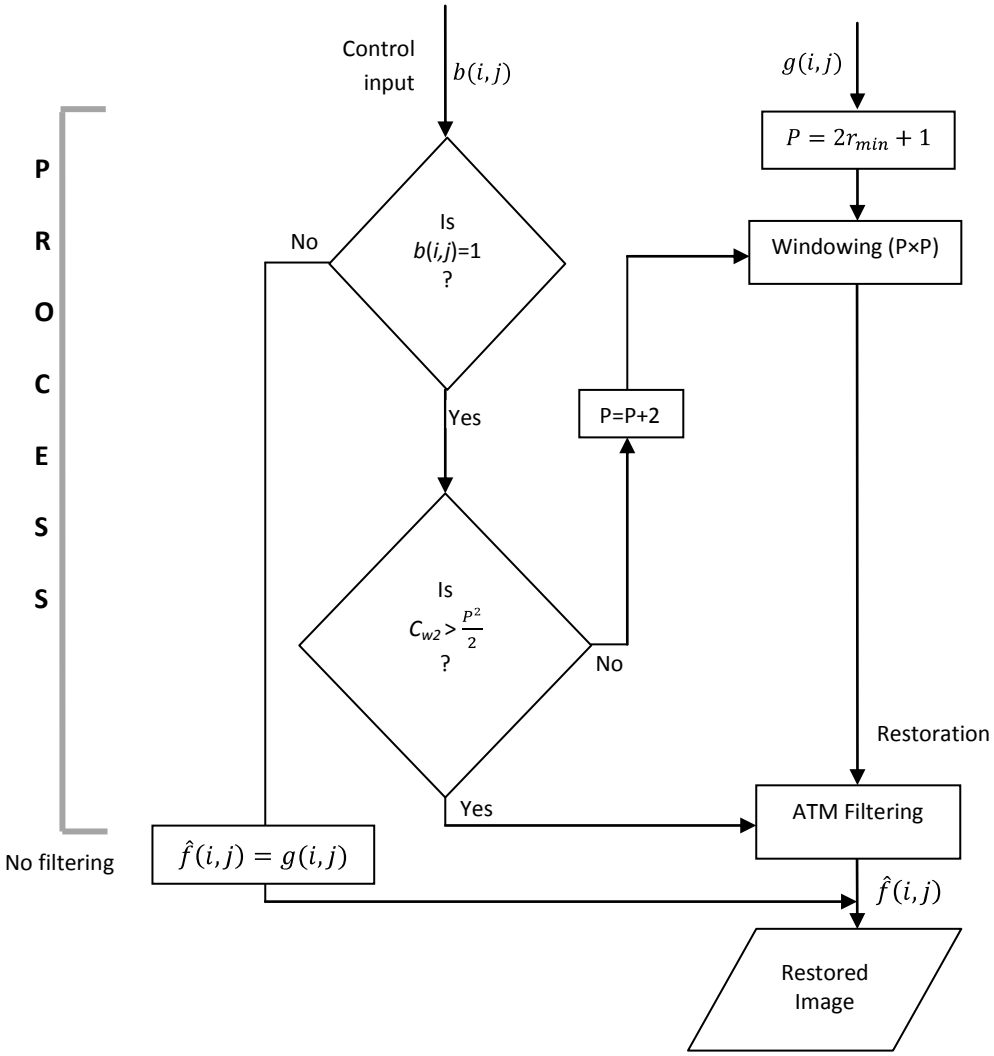


Fig. 3.12 Flowchart for restoration

This algorithm presents the best and simple technique to remove impulse noise from images at wide range of noise density. An advantage of this method is that it doesn't require any external threshold parameter; it is generated in the filtering window itself. Thus, no tuning or training is required. The simulation results are presented in Section-3.8.

3.8. Simulation Results

It is very important to test the performance of proposed algorithms. The simulations are carried out on a MATLAB-7.4 platform that sits over a Windows-XP operating system.

The performances of proposed filters are tested on difference test images. There are various standard test images which are used in literature for testing purpose. The test images employed here are *Lena, Boat,* and *Pepper*. All of them are 8-bit gray scale images of size 512×512.

Image metrics: PSNR, MSE, UQI and IEF and T_E are evaluated for performance-evaluation of filters.

The PSNR values of different filters are given in the tables: Table-3.3-Table-3.5. MSE values are tabulated in the tables: Table-3.6 through Table-3.8 whereas UQI values are shown in the tables: Table-3.9 through Table 3.11. Further the tables: Table-3.12 through Table 3.14 demonstrate the filters' performances in terms of IEF. Table-3.15 tabulates execution time of proposed and existing filters. The best results are highlighted for quick analysis in the tables.

The graphical representation of PSNR, MSE, UQI, and IEF of proposed filters and some high performing filters are illustrated in the figures: Fig. 3.13 through Fig. 3.16 for easy analysis.

For subjective evaluation, the output images of different filters are shown in the figures: Fig. 3.17 through Fig. 3.22.To show some samples of restored images, for subjective evaluation, only Lena and Pepper images are considered with 40%, 60% and 80% SPN noise densities.

Conclusions are drawn in the next section.

Table-3.3: Filtering performance of various filters in terms of PSNR (dB) for SPN
Test image: *Lena*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	33.74	27.28	21.75	17.38	14.12	11.59	9.6	7.94	6.54
2	MF [5×5]	31.44	30.67	29.11	26.22	20.32	16.75	12.88	9.66	7.29
3	MF [7×7]	29.41	28.99	28.49	27.66	25.91	22.06	16.54	11.80	8.09
4	MF [9×9]	28.13	27.86	27.53	26.97	26.25	24.88	19.91	14.00	8.97
5	MF [11×11]	27.18	27.02	26.76	26.38	25.83	25.15	22.32	16.21	9.90
6	MF [15×15]	26.37	26.23	26.02	25.27	25.41	24.66	23.44	18.45	10.64
7	MF [17×17]	27.71	25.53	25.41	25.02	24.98	24.35	23.46	20.33	11.74
8	ATM[3×3]	31.99	27.72	23.06	19.17	15.97	13.36	11.31	9.70	8.45
9	ATM[5×5]	29.11	27.61	25.96	23.89	20.68	17.33	14.57	12.48	10.92
10	ATM[7×7]	29.2	28.88	28.46	28.00	27.11	24.73	18.92	13.30	8.62
11	ATM[9×9]	25.96	24.55	23.45	22.47	21.62	20.50	17.26	14.56	12.78
12	ATM[11×11]	25.01	23.65	22.59	21.70	20.87	19.75	17.66	15.15	13.41
13	ATM[15×15]	24.42	22.83	21.81	20.99	20.17	19.28	17.49	15.28	13.60
14	ATM[17×17]	23.48	22.15	21.07	20.26	19.54	18.76	17.24	15.35	13.85
15	CWM	34.49	30.11	24.01	19.21	15.55	12.48	10.03	8.19	6.69
16	TSM	35.49	29.31	23.48	18.83	15.02	12.02	9.58	7.62	6.27
17	AMF	33.76	29.51	24.65	19.99	16.37	13.44	10.87	8.73	6.92
18	PSM	37.27	32.81	29.21	24.83	20.55	12.26	9.95	8.11	6.61
19	SMF	29.24	27.51	25.94	24.04	21.12	17.25	13.38	9.92	7.36
20	BDND	39.09	36.53	34.22	31.71	29.66	28.84	26.62	23.83	17.81
21	ANDS[P1]	**42.84**	35.91	30.05	24.16	20.11	16.63	13.42	10.28	7.62
22	REIR[P2]	41.48	38.80	37.22	33.03	31.11	29.45	27.48	25.68	23.68
23	IDPSM[P3]	40.78	37.44	34.14	30.90	29.14	27.55	26.14	24.78	21.74
24	IRIC[P4]	41.98	37.41	37.53	32.24	30.41	28.41	26.40	23.87	20.31
25	ASF-I[P5]	41.62	37.96	35.37	**33.89**	**32.48**	**30.93**	27.41	27.64	**25.35**
26	ASF-II[P6]	**42.96**	**39.05**	**36.34**	33.77	31.85	29.46	28.33	27.24	25.06
27	IASF[P7]	41.67	37.64	35.64	33.78	**32.45**	**31.01**	**29.28**	**27.75**	24.07

Table-3.4: Filtering performance of various filters in terms of PSNR (dB) for SPN
Test image: *Pepper*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	32.97	27.04	21.40	17.42	13.93	11.43	9.41	7,72	6.43
2	MF [5×5]	32.21	31.18	29.44	26.20	21.28	16.67	12.72	9.62	7.14
3	MF [7×7]	30.88	30.18	29.28	28.25	26.13	21.90	16.57	11.57	7.94
4	MF [9×9]	29.70	29.08	28.56	27.70	26.61	24.93	20.24	13.75	8.92
5	MF [11×11]	28.57	28.24	27.77	27.08	26.35	25.00	22.50	16.29	9.72
6	MF [15×15]	27.65	27.27	26.92	26.46	25.72	24.78	23.21	18.03	10.66
7	MF [17×17]	26.27	26.44	26.22	25.84	25.28	24.58	23.44	20.17	11.58
8	ATM[3×3]	31.99	27.72	23.03	19.17	15.97	13.36	11.31	9.70	8.45
9	ATM[5×5]	29.49	27.56	25.88	23.49	20.46	17.26	14.56	12.19	10.51
10	ATM[7×7]	29.20	28.88	28.46	28.00	27.11	24.73	18.92	13.30	8.62
11	ATM[9×9]	26.39	24.95	23.68	22.59	21.61	19.92	17.20	14.41	12.39
12	ATM[11×11]	26.31	24.58	23.30	22.30	21.40	20.20	17.88	15.17	13.44
13	ATM[15×15]	24.47	22.84	21.85	20.85	20.08	19.17	17.31	15.03	13.18
14	ATM[17×17]	23.69	22.23	21.11	20.15	19.40	18.54	17.01	1508	13.11
15	CWM	33.37	29.11	23.66	19.02	15.21	12.26	9.97	8.01	6.51
16	TSM	35.92	29.43	23.60	18.79	15.02	11.83	9.50	7.5	6.12
17	AMF	37.91	31.97	25.90	20.53	16.46	13.29	10.74	8.58	6.81
18	PSM	37.27	32.21	28.93	25.20	20.88	12.11	9.84	8.00	6.47
19	SMF	29.79	27.58	25.82	23.76	20.89	17.00	13.16	9.76	7.20
20	BDND	39.43	37.07	34.26	32.69	31.16	30.40	27.06	25.07	19.51
21	ANDS[P1]	**42.58**	35.41	29.59	24.02	20.07	16.57	13.44	10.21	7.54
22	REIR[P2]	41.17	37.24	34.28	32.52	31.01	29.02	27.41	25.53	23.26
23	IDPSM[P3]	38.35	35.40	32.74	30.82	29.22	27.82	25.72	23.94	21.73
24	IRIC[P4]	40.65	36.43	33.74	31.33	29.75	27.69	26.00	23.47	19.70
25	ASF-I[P5]	41.63	37.96	35.37	**33.89**	32.48	30.92	29.40	27.64	**25.34**
26	ASF-II[P6]	**42.43**	**38.77**	**36.21**	34.16	32.57	30.31	29.14	27.55	25.28
27	IASF[P7]	**42.00**	**38.34**	36.08	34.33	**32.86**	**31.25**	**29.63**	**28.22**	24.05

Table-3.5: Filtering performance of various filters in terms of PSNR (dB) for SPN
Test image: *Boat*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	29.86	25.80	21.20	17.21	14.00	11.59	9.53	7.92	6.59
2	MF [5×5]	27.33	26.78	25.82	24.01	20.39	16.30	12.68	9.71	7.3
3	MF [7×7]	25.38	25.13	24.81	23.32	23.25	20.54	16.10	11.63	8.06
4	MF [9×9]	24.15	24.02	23.80	23.46	23.15	22.17	19.05	13.50	8.97
5	MF [11×11]	29.58	29.27	28.85	28.41	27.78	26.57	23.50	16.35	10.01
6	MF [15×15]	22.84	22.74	22.67	22.54	22.34	21.93	21.18	17.49	10.80
7	MF [17×17]	22.51	22.41	22.37	22.25	22.07	21.79	21.27	19.05	11.73
8	ATM[3×3]	28.94	26.01	22.27	18.70	15.81	13.37	11.34	9.68	8.46
9	ATM[5×5]	26.16	25.16	24.10	22.42	19.85	17.12	14.44	12.35	10.83
10	ATM[7×7]	31.40	27.66	22.80	18.44	14.85	11.96	9.58	7.67	6.30
11	ATM[9×9]	23.13	22.43	21.75	21.14	20.47	19.07	16.82	14.51	12.77
12	ATM[11×11]	22.34	21.58	20.96	20.33	19.82	18.92	17.08	14.85	13.39
13	ATM[15×15]	21.76	21.01	20.33	19.71	19.21	18.55	17.07	15.12	13.51
14	ATM[17×17]	21.31	20.54	19.90	19.25	18.70	18.10	16.80	15.12	13.68
15	CWM	25.17	24.91	24.76	24.46	23.90	22.53	18.42	13.03	8.56
16	TSM	31.42	27.72	22.95	18.67	14.99	11.94	9.57	7.68	6.29
17	AMF	33.76	29.51	24.65	19.99	16.36	13.44	10.87	8.75	6.92
18	PSM	32.06	29.49	26.61	23.74	20.28	16.29	9.94	8.14	6.66
19	SMF	26.19	25.11	24.03	22.60	20.45	16.93	13.22	9.87	7.33
20	BDND	33.86	32.35	30.31	28.52	26.94	25.26	23.29	22.03	17.85
21	ANDS[P1]	**38.32**	33.33	29.32	23.29	19.59	16.09	13.10	10.00	7.45
22	REIR[P2]	35.33	32.54	30.66	28.88	27.48	25.72	24.21	22.56	20.74
23	IDPSM[P3]	33.15	31.27	28.95	26.50	25.17	24.05	23.09	21.95	19.76
24	IRIC[P4]	37.72	33.46	30.90	29.01	27.25	25.68	24.13	22.47	20.47
25	ASF-I[P5]	37.30	33.67	31.62	**30.04**	**28.78**	**27.44**	25.98	24.52	**22.83**
26	ASF-II[P6]	**38.54**	**34.90**	**32.16**	**30.01**	28.03	25.97	24.95	24.08	22.37
27	IASF[P7]	37.36	33.68	31.60	**30.01**	**28.63**	27.32	**26.05**	**24.62**	22.03

Table-3.6: Filtering performance of various filters in terms of MSE for SPN
Test image: *Lena*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	27.43	121	432	1180	2510	4510	7130	10400	14421
2	MF [5×5]	46.66	56	80	155	480	1383	33505	7017	12156
3	MF [7×7]	74.48	82	92	111	168	405	1440	4290	8080
4	MF [9×9]	99.79	106.2	114.7	130.5	153.9	210.9	663.1	2584	8230
5	MF [11×11]	124.4	128.8	136.9	149.4	169.6	198.4	381.1	1554	6649
6	MF [15×15]	149.6	154.6	162.5	171.1	186.8	222.1	294.2	927.3	5610
7	MF [17×17]	174.4	181.9	187.0	196.2	206.1	238.5	292.5	601.4	4355
8	ATM[3×3]	41.08	109.9	320.8	786.3	1644	2994	4805	6960	9281
9	ATM[5×5]	79.7	112.6	164.6	265.4	554.9	1184	2265	3670	5261
10	ATM[7×7]	17.73	71.37	264	835	2045	7009	7120	11090	15076
11	ATM[9×9]	164.9	228.0	293.1	367.9	447.1	638.3	1221	2278	3415
12	ATM[11×11]	204.7	280.1	357.6	439.4	531.9	687.7	1113	1984	2961
13	ATM[15×15]	251.2	338.2	428.2	517.5	623.9	767.0	1157	1926	2833
14	ATM[17×17]	291.2	395.2	507.8	611.9	711.8	663.8	1227	1896	2677
15	CWM	23	63	259	789	1803	3661	6450	9855	13967
16	TSM	18.33	76.16	291.7	850.3	2047	4066	7162	11235	15324
17	AMF	27.30	72.76	222.8	650.8	1502	2943	5316	8702	13196
18	PSM	12.17	33.98	78.18	209	571	3856	6573	10035	14164
19	SMF	77.38	115.3	165.5	255.8	501.5	1223	2982	6609	11934
20	BDND	7.03	13.10	18.50	30.67	48.32	67.10	129.2	155.1	698.8
21	ANDS[P1]	**3.37**	16.66	64.25	249	648	1412	2955	6096	11236
22	REIR[P2]	4.56	9.13	16.47	29	46.34	73.1	116	184	299
23	IDPSM[P3]	5.43	11.71	25.00	52.9	79.26	114.4	158.2	216.5	895.5
24	IRIC[P4]	4.12	11.77	22.92	38.80	59.15	93.77	148.9	266.7	604.8
25	ASF-I[P5]	4.47	10.40	18.90	**26.54**	**36.71**	52.5	76.56	111.8	**189**
26	ASF-II[P6]	**3.28**	**8.08**	**15.09**	27.32	42.48	73.57	95.6	112.5	203
27	IASF[P7]	4.41	11.19	17.74	27.23	**36.97**	**51.68**	**74.68**	**108.9**	255.2

Table-3.7: Filtering performance of various filters in terms of MSE for SPN
Test image: *Pepper*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	47	120	352	841	1724	3082	4973	7325	9658
2	MF [5×5]	36	49.5	74	156	485	1408	3484	7098	12660
3	MF [7×7]	74.4	82	92	111	167	404	1445	4290	8086
4	MF [9×9]	69.61	80.24	90.42	110.2	141.9	208.9	615.2	2738	8325
5	MF [11×11]	90.28	97.30	108.5	127.2	150.4	205.5	365.2	152	6928
6	MF [15×15]	111.5	121.6	131.8	146.4	173.9	215.9	310.1	1023	5578
7	MF [17×17]	137.6	147.5	155.1	169.1	192.2	226.3	294.4	626.6	4537
8	ATM[3×3]	33.8	124	462	1201	2571	4696	10720	14700	14864
9	ATM[5×5]	73	114	168	292	584	1221	2275	3921	5773
10	ATM[7×7]	58.49	62.57	74.5	91	115	205	797	3220	9052
11	ATM [9×9]	149.2	207.7	278.2	357.7	448.6	661.8	1238	2350	3743
12	ATM[11×11]	190.6	274.8	354.8	445.5	548.5	689.5	1194	2154	3423
13	ATM[15×15]	232.1	337.6	427.1	533.9	638.3	786.8	1207	2040	3126
14	ATM[17×17]	277.7	388.4	503.2	626.8	745.1	908.7	1292	2015	3174
15	CWM	29.9	79.7	280	815	1950	3863	6581	10245	14509
16	TSM	16.6	74.1	284	860	2043	4266	7281	11481	15870
17	AMF	10.5	41.2	167	575	1466	3044	5482	9014	13544
18	PSM	12.1	39	84	196	531	3997	6738	10283	14650
19	SMF	68.2	114	170	274	529	1297	3138	6859	12372
20	BDND	6.42	10.12	15.3	26	40	60	81	128	724
21	ANDS[P1]	**3.5**	18.6	71	257	639	1429	2943	6182	11448
22	REIR[P2]	10.18	16	27.3	39.85	61	91.7	137	198	306
23	IDPSM[P3]	9.9	21.6	41	65	90	131	175	263	979
24	IRIC[P4]	5.5	14.8	28	48	68	110	163	293	696
25	ASF-I[P5]	4.06	9.6	**15.7**	**23.3**	34	**45.4**	**68.4**	**98.6**	**175**
26	ASF-II[P6]	3.7	**8.6**	**15.5**	25	36	60	79	114	193
27	IASF[P7]	4.09	9.5	16	24	**33.6**	49	71	**98**	251

Table-3.8: Filtering performance of various filters in terms of MSE for SPN
Test image: *Boat*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	67	171	494	1247	2580	4513	7239	10538	14368
2	MF [5×5]	120	136	170	258	504	1524	3508	6930	12059
3	MF [7×7]	188	199	215	302	308	574	1597	4452	10140
4	MF [9×9]	249.6	257.6	270.6	292.5	314.9	394.5	808.9	2902	8323
5	MF [11×11]	71.5	76.77	84.58	93.56	108.3	143.1	290.3	1506	6481
6	MF [15×15]	337.7	345.5	351.3	361.9	378.6	416.3	494.5	1157	5399
7	MF [17×17]	364.5	373.1	375.9	386.5	403.5	430.3	485.1	808.4	4364
8	ATM[3×3]	82	163	385	877	1707	2990	4767	6988	9254
9	ATM[5×5]	157	198	253	372	672	1259	2336	3780	5369
10	ATM[7×7]	198	210	217	233	265	363	934	3231	9057
11	ATM [9×9]	316.1	371.1	434.5	500.0	582.2	804.3	1351	2299	3431
12	ATM[11×11]	379.1	451.2	521.2	602.1	676.6	833.4	1272	2126	2972
13	ATM[15×15]	432.9	514.2	601.8	795.8	779.4	906.4	1272	1997	2896
14	ATM[17×17]	480.5	574.2	665.1	772.4	876.1	1006	1358	1997	2784
15	CWM	67	124	342	887	1928	3753	6368	9825	13947
16	TSM	47	109	330	890	2061	4158	7183	11091	15260
17	AMF	27.3	73	222	650	1501	2942	5315	8702	13196
18	PSM	40	73	141	274	609	1527	6599	9988	14008
19	SMF	156	200	257	357	584	1318	3095	6691	12080
20	BDND	17	30	48	73	104	153	191	256	845
21	ANDS[P1]	**9.5**	30	96	304	714	1598	3180	6501	11692
22	REIR[P2]	24	36	56	84	116	174	246	358	548
23	IDPSM[P3]	31.4	49	83	145	197	256	318	415	1345
24	IRIC[P4]	11	30	53	84	122	175	250	368	582
25	ASF-I[P5]	12	28	45	**65**	**86**	**117**	165	225	**340**
26	ASF-II[P6]	**9**	**21**	**40**	66	102	164	208	253	376
27	IASF[P7]	11.9	28	45	**65**	90	120	**162**	**224**	416

Table-3.9: Filtering performance of various filters in terms of UQI for SPN
Test image: *Lena*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	0.994	0.973	0.907	0.788	0.612	0.432	0.268	0.153	0.068
2	MF [5×5]	0.994	0.973	0.910	0.780	0.605	0.428	0.279	0 .161	0.061
3	MF [7×7]	0.983	0.981	0.979	0.975	0.963	0.913	0.735	0.437	0.158
4	MF [9×9]	0.977	0.976	0.974	0.970	0.965	0.953	0.861	0.591	0.220
5	MF [11×11]	0.972	0.971	0.969	0.966	0.961	0.955	0.916	0.715	0.285
6	MF [15×15]	0.966	0.964	0.963	0.961	0.957	0.949	0.933	0.809	0.337
7	MF [17×17]	0.960	0.958	0.957	0.955	0.952	0.945	0.933	0.869	0.412
8	ATM[3×3]	0.990	0.976	0.931	0.840	0.698	0.522	0.352	0.204	0.096
9	ATM[5×5]	0.982	0.975	0.963	0.941	0.880	0.753	0.568	0.371	0.177
10	ATM[7×7]	0.987	0.986	0.985	0.984	0.981	0.976	0.958	0.901	0.767
11	ATM [9×9]	0.963	0.949	0.935	0.919	0.902	0.862	0.737	0.513	0.277
12	ATM[11×11]	0.953	0.937	0.921	0.904	0.884	0.850	0.758	0.561	0.306
13	ATM[15×15]	0.943	0.924	0.905	0.886	0.865	0.835	0.748	0.563	0.306
14	ATM[17×17]	0.933	0.911	0.888	0.866	0.844	0.813	0.731	0.561	0.316
15	CWM	0.996	0.993	0.986	0.972	0.946	0.901	0.831	0.745	0.657
16	TSM	0.995	0.983	0.938	0.835	0.662	0.466	0.274	0.132	0.050
17	AMF	0.993	0.983	0.950	0.863	0.722	0.550	0.364	0.201	0.076
18	PSM	0.997	0.992	0.983	0.956	0.890	0.501	0.326	0.192	0.078
19	SMF	0.982	0.974	0.964	0.944	0.896	0.771	0.556	0.298	0.120
20	BDND	0.999	**0.9981**	0.9962	0.993	0.990	0.985	0.980	0.970	0.863
21	ANDS[P1]	**0.9992**	0.996	0.986	0.947	0.872	0.753	0.569	0.337	0.120
22	REIR[P2]	0.998	0.997	0.994	0.992	0.987	0.981	0.971	0.958	0.933
23	IDPSM[P3]	0.998	0.997	0.994	0.988	0.982	0.974	0.964	0.952	0.792
24	IRIC[P4]	0.9990	0.997	0.994	0.991	0.987	0.979	0.967	0.941	0.867
25	ASF-I[P5]	0.9990	0.997	0.995	**0.9941**	**0.9919**	0.9884	**0.9836**	0.9754	**0.9581**
26	ASF-II[P6]	**0.9993**	**0.9982**	**0.9967**	0.9940	0.990	0.983	0.978	0.972	0.954
27	IASF[P7]	0.9990	0.997	0.9961	0.9940	**0.9919**	0**.9886**	0.9831	**0.9760**	0.9445

Table-3.10: Filtering performance of various filters in terms of UQI for SPN
Test image: *Pepper*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	20	30	40	50	60	70	80	90
1	MF [3×3]	0.994	0.978	0.922	0.814	0.651	0.479	0.322	0.185	0.087
2	MF [5×5]	0.993	0.991	0.987	0.973	0.919	0.789	0.571	0.338	0.135
3	MF [7×7]	0.990	0.989	0.986	0.983	0.972	0.929	0.784	0.485	0.194
4	MF [9×9]	0.987	0.985	0.984	0.980	0.975	0.963	0.878	0.633	0.282
5	MF [11×11]	0.984	0.982	0.980	0.977	0.973	0.963	0.936	0.766	0.319
6	MF [15×15]	0.980	0.978	0.976	0.973	0.969	0.961	0.945	0.832	0.408
7	MF [17×17]	0.975	0.973	0.972	0.969	0.965	0.959	0.946	0.892	0.471
8	ATM[3×3]	0.991	0.979	0.940	0.862	0.736	0.574	0.400	0.241	0.122
9	ATM[5×5]	0.987	0.980	0.970	0.949	0.899	0.793	0.633	0.407	0.191
10	ATM[7×7]	0.992	0.990	0.990	0.988	0.985	0.981	0.962	0.894	0.757
11	ATM [9×9]	0.973	0.963	0.951	0.938	0.922	0.884	0.784	0.581	0.318
12	ATM[11×11]	0.966	0.951	938	0.922	0.905	0.880	0.788	0.595	0.312
13	ATM[15×15]	0.958	0.940	0.925	0.907	0.889	0.863	0.783	0.613	0.345
14	ATM[17×17]	0.950	0.931	0.911	0.890	0.870	0.842	0.769	0.603	0.314
15	CWM	0.995	0.993	0.986	0.971	0.940	0.891	0.822	0.730	0.625
16	TSM	0.997	0.987	0.952	0.865	0.717	0.512	0.330	0.170	0.060
17	AMF	0.998	0.992	0.971	0.906	0.783	0.612	0.427	0.253	0.113
18	PSM	0.997	0.993	0.985	0.967	0.915	0.541	0.367	0.218	0.088
19	SMF	0.988	0.980	0.970	0.953	0.912	0.803	0.599	0.343	0.144
20	BDND	0.998	0.997	0.996	0.995	0.993	0.989	0.985	0.977	0.880
21	ANDS[P1]	**0.9994**	0.996	0.987	0.957	0.898	0.791	0.626	0.380	0.135
22	REIR[P2]	0.998	0.997	0.995	0.992	0.990	0.984	0.977	0.964	0.938
23	IDPSM[P3]	0.997	0.995	0.992	0.988	0.982	0.976	0.969	0.954	0.927
24	IRIC[P4]	0.9990	0.997	0.995	0.991	0.988	0.980	0.971	0.949	0.880
25	ASF-I[P5]	0.9991	0.998	0.9972	**0.9958**	**0.9942**	**0.9921**	**0.9881**	**0.9829**	**0.9694**
26	ASF-II[P6]	**0.9994**	**0.9985**	**0.9973**	0.9956	0.993	0.989	0.986	0.980	0.965
27	IASF[P7]	0.9990	0.9983	0.9972	0.9957	0.9941	0.991	0.9877	**0.9829**	0.956

Table-3.11: Filtering performance of various filters in terms of UQI for SPN
Test image: *Boat*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
1	MF [3×3]	0.984	0.960	0.892	0.760	0.581	0.411	0.256	0.145	0.061
2	MF [5×5]	0.971	0.967	0.959	0.940	0.869	0.709	0.481	0.267	0.102
3	MF [7×7]	0.955	0.952	0.948	0.928	0.927	0.870	0.693	0.400	0.143
4	MF [9×9]	0.940	0.938	0.934	0.929	0.924	0.906	0.820	0.512	0.195
5	MF [11×11]	0.982	0.981	0.979	0.977	0.973	0.965	0.931	0.704	0.283
6	MF [15×15]	0.918	0.916	0.914	0.911	0.907	0.899	0.881	0.750	0.318
7	MF [17×17]	0.911	0.908	0.908	0.905	0.901	0.894	0.881	0.811	0.369
8	ATM[3×3]	0.980	0.962	0.913	0.814	0.675	0.508	0.341	0.193	0.086
9	ATM[5×5]	0.9962	0.953	0.940	0.913	0.846	0.727	0.540	0.332	0.135
10	ATM[7×7]	0.973	0.972	0.971	0.970	0.966	0.960	0.944	0.899	0.792
11	ATM [9×9]	0.923	0.911	0.897	0.882	0.864	0.813	0.694	0.488	0.256
12	ATM[11×11]	0.908	0.892	0.876	0.859	0.842	0.807	0.712	0.506	0.292
13	ATM[15×15]	0.894	0.876	0.857	0.838	0.819	0.792	0.706	0.522	0.261
14	ATM[17×17]	0.882	0.862	0.843	0.820	0.798	0.769	0.681	0.509	0.265
15	CWM	0.991	0.987	0.981	0.968	0.946	0.912	0.864	0.797	0.722
16	TSM	0.989	0.974	0.927	0.820	0.645	0.439	0.264	0.136	0.049
17	AMF	0.993	0.983	0.950	0.863	0.722	0.550	0.364	0.201	0.076
18	PSM	0.990	0.983	0.968	0.940	0.876	0.732	0.304	0.171	0.075
19	SMF	0.963	0.953	0.940	0.918	0.871	0.741	0.523	0.280	0.099
20	BDND	0.996	0.993	0.988	0.983	0.975	0.963	0.954	0.939	0.817
21	ANDS[P1]	0.9977	0.993	0.978	0.932	0.853	0.712	0.531	0.302	0.113
22	REIR[P2]	0.994	0.991	0.987	0.980	0.973	0.960	0.944	0.920	0.879
23	IDPSM[P3]	0.992	0.988	0.980	0.966	0.953	0.939	0.925	0.903	0.732
24	IRIC[P4]	0.997	0.993	0.987	0.980	0.971	0.959	0.942	0.915	0.865
25	ASF-I[P5]	0.997	0.993	0.989	**0.9849**	**0.9798**	**0.9726**	0.9610	0.9465	**0.9186**
26	ASF-II[P6]	**0.9979**	**0.9951**	**0.9908**	0.9846	0.976	0.961	0.950	0.939	0.908
27	IASF[P7]	0.997	0.993	0.989	**0.9849**	0.9792	0.9718	**0.9620**	**0.9468**	0.903

Table-3.12: Filtering performance of various filters in terms of IEF for SPN
Test image: *Lena*

Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
1	MF [3×3]	68.38	30.47	12.86	6.25	3.70	2.47	1.82	1.42	1.16
2	MF [5×5]	40.07	65.85	71.29	48.45	20.87	8.25	3.75	2.12	1.38
3	MF [7×7]	24.93	44.71	59.73	66.68	55.71	27.49	9.00	3.46	1.65
4	MF[9×9]	18.48	34.86	48.58	56.85	60.13	52.73	19.63	5.74	2.07
5	MF[11×11]	14.83	28.75	40.67	49.67	54.58	56.05	34.16	9.55	2.51
6	MF [15×15]	12.46	23.99	34.32	43.09	49.49	49.99	44.21	15.98	2.97
7	MF [17×17]	10.61	20.30	29.89	37.90	45.10	46.72	44.26	24.61	3.83
8	ATM[3×3]	45.14	33.89	17.3	9.45	5.62	3.72	2.70	2.13	1.79
9	ATM[5×5]	23.34	32.86	33.82	28.09	16.74	9.42	5.73	4.04	3.17
10	ATM[7×7]	23.76	43.67	59.49	72.34	73.23	50.76	15.60	4.88	1.87
11	ATM[9×9]	15.01	20.39	22.25	23.53	23.85	19.12	10.98	6.94	4.95
12	ATM[11×11]	9.02	13.25	15.49	16.82	17.48	16.21	11.62	7.46	5.63
13	ATM[15×15]	7.38	10.85	12.85	14.37	14.88	14.47	11.21	7.69	5.90
14	ATM[17×17]	6.38	9.35	10.97	12.18	12.85	12.90	10.58	7.81	6.23
15	CWM	79.78	58.42	21.59	9.53	5.10	3.03	2.01	1.50	1.19
16	TSM	100.7	48.85	19.1	8.72	4.53	2.73	1.81	1.32	1.08
17	AMF	67.25	50.83	24.87	11.34	6.15	3.74	2.42	1.69	1.26
18	PSM	151.8	109.1	71.36	35.55	16.28	2.88	1.97	1.47	1.18
19	SMF	23.88	32.37	33.79	29.01	18.43	9.11	4.34	2.24	1.39
20	BDND	465.3	407.9	358.2	266.4	209.2	166.1	145.5	109.8	25.06
21	ANDS[P1]	552.9	222.7	86.45	29.60	14.32	7.89	4.39	2.42	1.48
22	REIR[P2]	233.4	269.8	239.3	207.5	160.3	127.1	99.22	74.79	52.31
23	IDPSM[P3]	341.9	313.4	219.5	140.5	116.8	97.10	82.09	68.50	15.26
24	IRIC[P4]	450.9	316.0	242.2	191.3	157.0	119.0	87.28	55.56	27.57
25	ASF-I[P5]	411.6	357.9	205.8	**279.7**	**253.7**	211.9	**173.8**	132.7	**88.03**
26	ASF-II[P6]	**567.7**	**462.1**	**365.4**	271.1	218.5	151.8	135.9	121.0	82.59
27	IASF[P7]	420.1	330.8	313.9	273.5	251.2	**215.7**	169.7	**136.1**	65.40

Table-3.13: Filtering performance of various filters in terms of IEF for SPN
Test image: *Pepper*

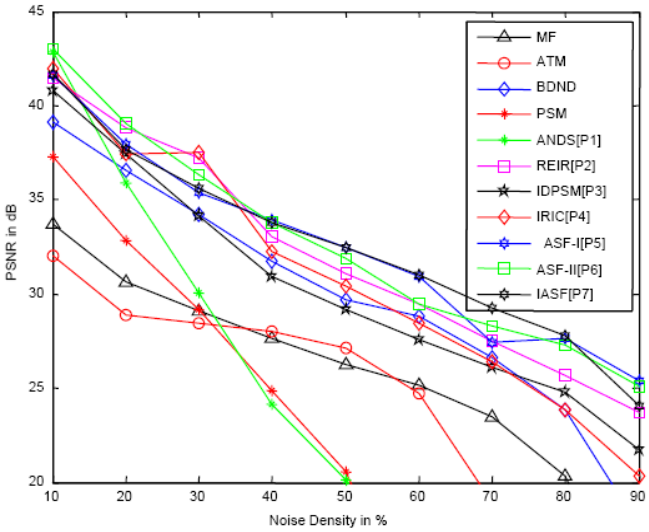
Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
1	MF [3×3]	57.91	39.86	12.3	6.27	3.66	2.47	1.81	1.41	1.16
2	MF [5×5]	50.22	77.37	77.06	48.92	20.86	7.87	3.87	2.14	1.38
3	MF [7×7]	36.5	61.58	75.48	79.41	60.90	27.48	9.40	3.42	1.65
4	MF[9×9]	27.72	47.39	63.86	69.68	67.69	55.33	21.88	5.60	2.06
5	MF[11×11]	21.25	39.26	53.04	60.72	63.76	56.14	36.81	10.06	2.50
6	MF [15×15]	17.50	31.82	43.97	52.67	55.16	53.54	43.43	15.02	3.09
7	MF [17×17]	14.02	26.25	37.21	45.52	49.84	50.94	45.55	24.64	3.80
8	ATM[3×3]	41.03	31.84	16.41	9.41	5.57	3.75	2.70	2.09	1.78
9	ATM[5×5]	26.39	33.73	34.47	26.49	16.49	9.42	5.88	3.92	2.99
10	ATM[7×7]	32.83	57.27	77.36	84.53	83.23	56.16	16.88	4.64	1.90
11	ATM[9×9]	12.82	18.47	20.62	21.51	21.43	17.47	10.87	6.53	4.61
12	ATM[11×11]	10.04	14.00	16.25	17.28	17.57	16.75	11.26	7.13	5.05
13	ATM[15×15]	8.32	11.34	13.61	14.37	15.05	14.68	11.20	7.53	5.52
14	ATM[17×17]	6.96	9.93	11.45	12.29	12.96	12.74	10.41	7.61	5.45
15	CWM	64.47	48.52	20.66	9.42	4.91	2.98	2.03	1.50	1.19
16	TSM	115.2	51.85	20.32	8.91	4.68	2.71	1.84	1.33	1.09
17	AMF	181	92.61	34.19	13.34	6.56	3.78	2.44	1.70	1.27
18	PSM	158.2	98.03	69.39	39.08	18.06	2.89	1.99	1.49	1.18
19	SMF	28.02	34.00	34.18	28	18.22	8.88	4.28	2.24	1.39
20	BDND	409.6	381	375.1	296.7	243	195	167	120.3	24
21	ANDS[P1]	**533.2**	206.8	80.62	29.90	15.02	8.08	4.57	2.48	1.52
22	REIR[P2]	178.9	229.1	212	183.3	168.3	127.1	100.3	73.45	46.44
23	IDPSM[P3]	162	162.2	131.5	113.2	98	85.1	77.20	58.66	15.71
24	IRIC[P4]	343.2	255.9	208.8	160.4	140.1	104.3	82.4	52.48	24.77
25	ASF-I[P5]	473.8	400.5	366	315.3	**286.6**	**253.8**	**196**	**156.2**	**99.13**
26	ASF-II[P6]	513.7	**446.2**	**370**	308.7	266.6	190.6	169.1	134.7	89.62
27	IASF[P7]	468.7	405.5	360	**320.6**	284.1	235.5	190.1	**157.5**	67.64

Table-3.14: Filtering performance of various filters in terms of IEF for SPN
Test image: *Boat*

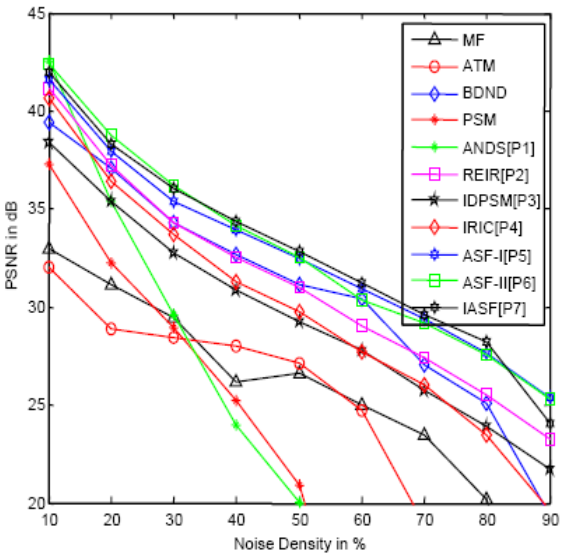
Sl. No	Filters	% of Noise (Salt-and-Pepper)								
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**
1	MF [3×3]	27.46	21.62	11.21	5.97	3.56	2.44	1.79	1.40	1.16
2	MF [5×5]	15.31	26.92	32.34	28.47	15.55	7.26	3.69	2.12	1.37
3	MF [7×7]	9.69	18.55	25.74	30.6	30.07	19.32	8.09	3.3	1.63
4	MF[9×9]	7.37	14.28	20.41	25.30	29.28	27.90	15.98	5.09	1.99
5	MF[11×11]	26.08	47.94	65.56	78.79	85.31	77.37	44.48	9.81	2.55
6	MF [15×15]	5.43	10.71	15.79	20.38	24.32	26.51	26.06	12.74	3.07
7	MF [17×17]	5.03	9.91	14.74	19.09	22.89	25.63	26.57	18.24	3.81
8	ATM[3×3]	22.17	22.62	14.32	8.45	5.39	3.38	2.78	2.11	1.79
9	ATM[5×5]	11.67	18.67	21.96	19.82	13.73	8.75	5.51	3.90	3.10
10	ATM[7×7]	9.28	17.65	25.53	31.71	34.94	30.41	13.8	4.56	1.84
11	ATM[9×9]	5.80	9.93	12.74	14.71	15.84	13.84	9.53	6.41	4.83
12	ATM[11×11]	4.85	8.20	10.57	12.27	13.63	13.27	10.10	6.93	5.57
13	ATM[15×15]	4.23	7.18	9.25	10.61	11.85	12.19	10.12	7.39	5.74
14	ATM[17×17]	3.81	6.43	8.25	9.53	10.50	11.01	9.53	7.38	5.96
15	CWM	27.77	30	16,12	8.42	4.78	2.94	2.02	1.5	1.19
16	TSM	39.32	33.63	16.70	8.25	4.48	2.66	1.70	1.33	1.08
17	AMF	67.27	50.82	24.87	11.34	6.15	3.74	2.42	1.69	1.26
18	PSM	46.13	50.32	39.05	26.80	15.12	7.22	1.95	1.47	1.18
19	NSMF	11.73	18.49	21.58	20.67	15.74	8.37	4.16	2.20	1.37
20	BDND	109.8	123.2	115.1	101	88.51	72.35	67.32	57.28	19.57
21	ANDS[P1]	193.1	122.6	57.80	24.13	12.91	6.94	4.06	2.26	1.41
22	REIR[P2]	76.92	101.6	98.94	87.98	79.26	63.08	52.43	41.18	30.32
23	IDPSM[P3]	58.93	76.98	66.61	50.32	46.70	43.03	40.50	35.52	12.16
24	IRIC[P4]	167.2	125.9	105.8	87.35	75.58	63.40	51.55	40.07	28.41
25	ASF-I[P5]	150.36	131.67	123.6	**114.1**	**106.7**	**94.77**	78.41	**65.34**	**48.67**
26	ASF-II[P6]	**202.3**	**175.6**	**140.1**	111.6	90	67.6	62.1	58.11	44.13
27	IASF[P7]	153.1	132.7	123.6	**113.9**	104.4	91.50	**80.1**	**65.80**	40.01

Table-3.15: Execution time (seconds), T_E taken by various filters for Lena image at 10% noise density

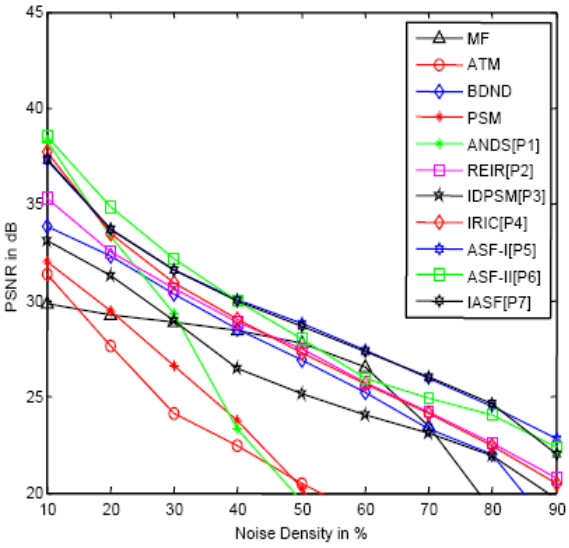
Sl. No.	Filters	*Execution time (seconds) in three different hardware platforms*		
		SYSTEM-1	SYSTEM-2	SYSTEM-3
1	MF [3×3]	22.13	7.27	8.22
2	MF [5×5]	22.18	7.24	8.23
3	MF [7×7]	22.35	7.41	8.27
4	MF[9×9]	22.60	7.52	8.30
5	MF[11×11]	23.04	7.68	8.32
6	MF [15×15]	24.23	8.38	8.50
7	MF [17×17]	25.27	8.97	8.55
8	ATM[3×3]	7.64	2.71	2.43
9	ATM[5×5]	8.32	2.78	2.45
10	ATM[7×7]	8.76	3.04	2.58
11	ATM[9×9]	9.4	3.33	2.68
12	ATM[11×11]	10.55	3.74	2.77
13	ATM[15×15]	14.38	4.74	2.89
14	ATM[17×17]	15.41	5.24	3.01
15	CWM	20.51	6.74	7.44
16	TSM	19.62	6.23	6.45
17	AMF	4.84	1.35	3.36
18	PSM	3.00	1.74	1.72
19	NSMF	14.95	5.14	4.75
20	BDND	24.62	7.92	8.60
21	ANDS[P1]	22.14	7.82	7.20
22	REIR[P2]	42.29	14.30	15.20
23	IDPSM[P3]	77.66	25.01	27.12
24	IRIC[P4]	6.87	2.15	2.14
25	ASF-I[P5]	2.87	1.02	1.08
26	**ASF-II[P6]**	**2.47**	**0.85**	**1.01**
27	IASF[P7]	17.7	6.02	5.60



(a)



(b)



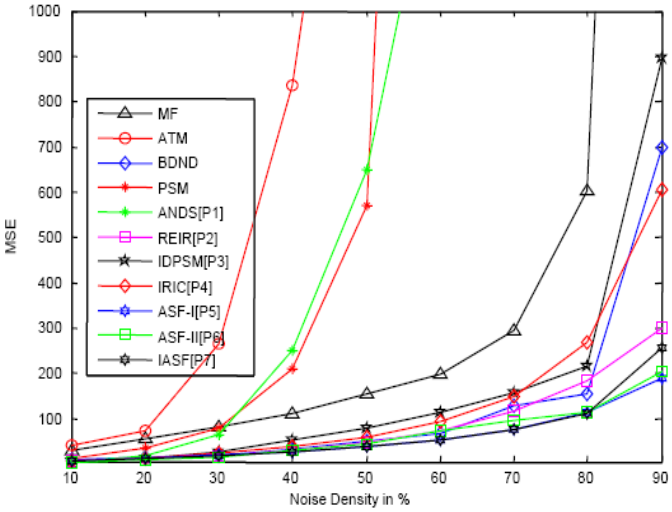
(c)

Fig. 3.13 Performance of various filters in terms of PSNR (dB) for SPN at different noise densities on the images:

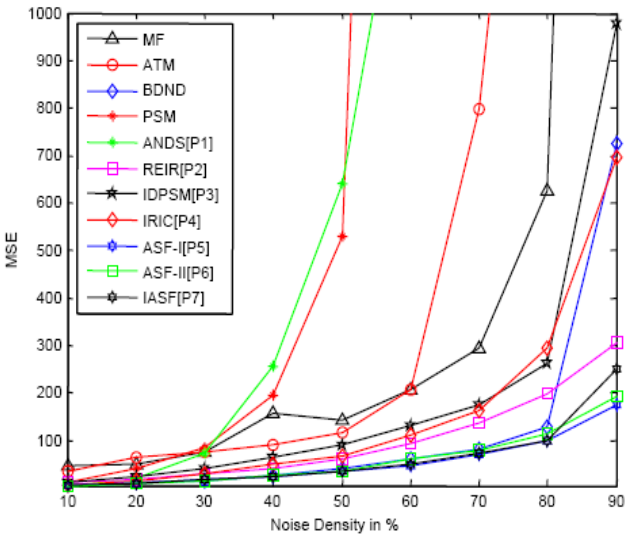
(a) Lena

(b) Pepper

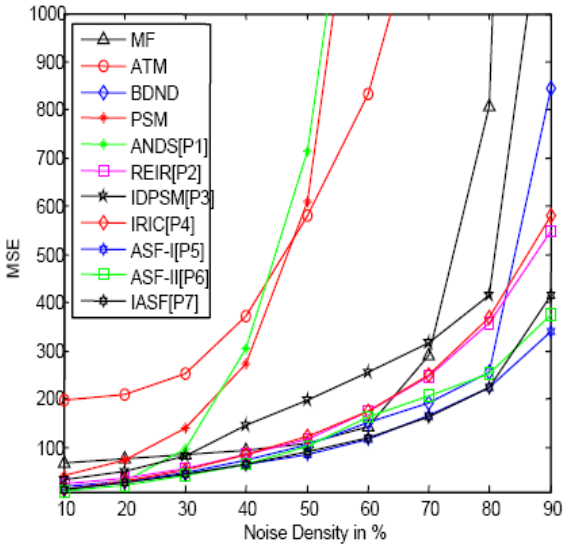
(c) Boat



(a)



(b)



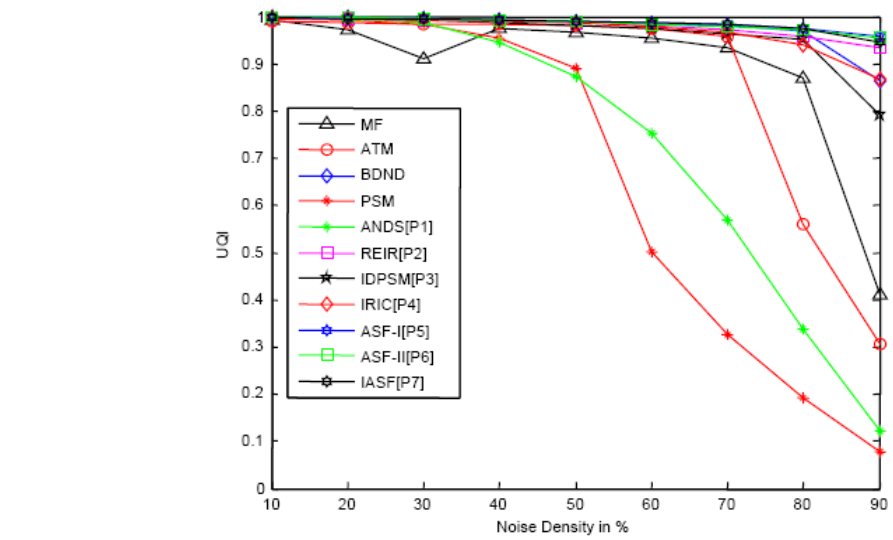
(c)

Fig. 3.14 Performance comparison of various filters in terms of MSE for SPN at different noise densities on the images:

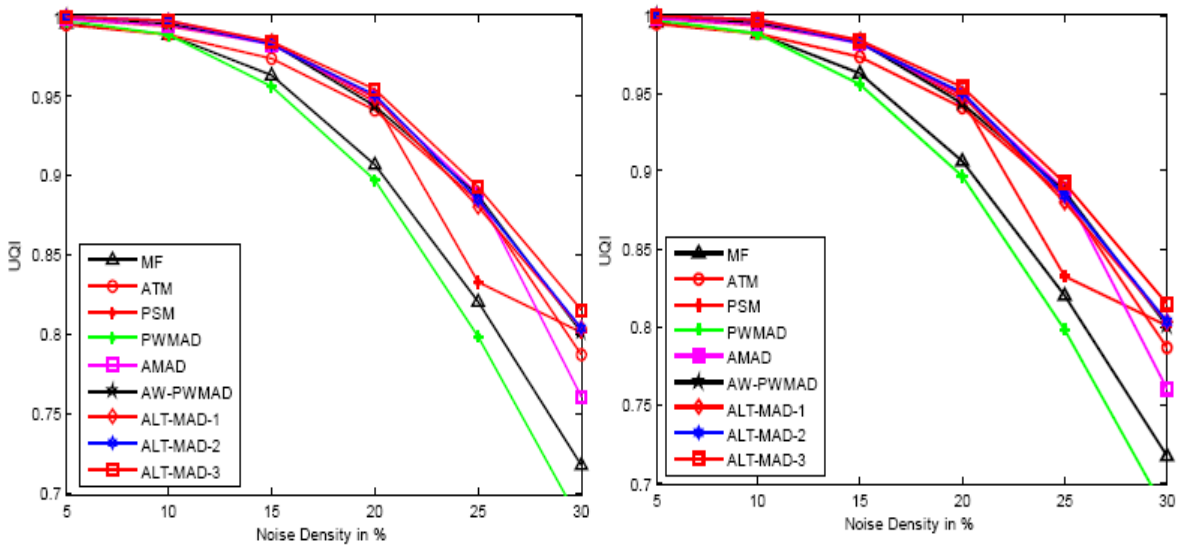
(a) Lena

(b) Pepper

(c) Boat



(a)



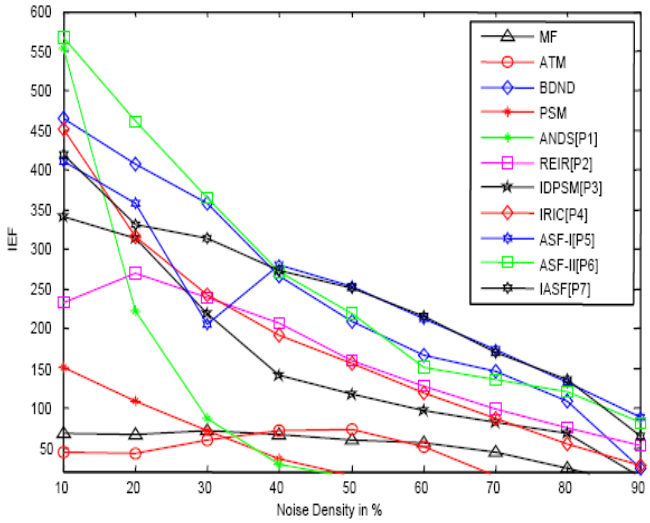
(b) (c)

Fig. 3.15 Performance comparison of various filters in terms of UQI for SPN at different noise densities on the images:

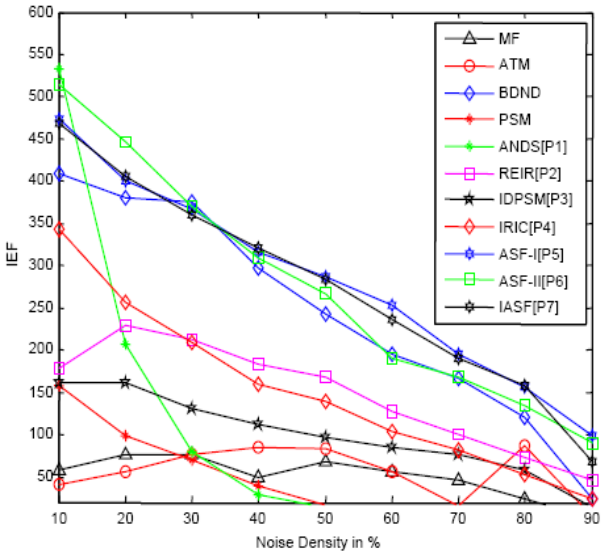
(a) Lena

(b) Pepper

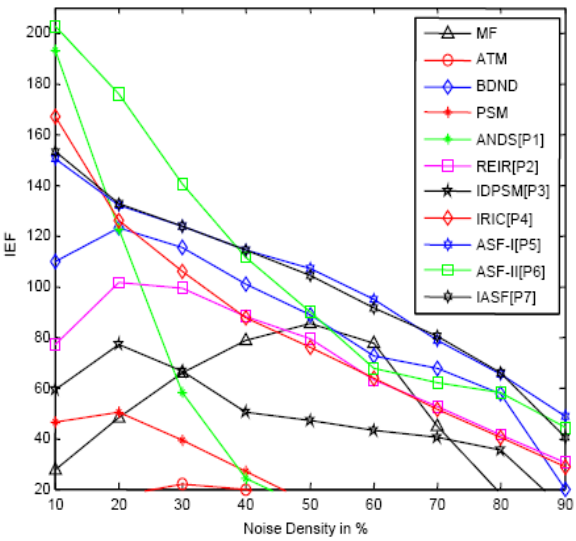
(c) Boat



(a)



(b)



(c)

Fig. 3.16 Performance comparison of various filters in terms of IEF for SPN at different noise densities on the images:

(a) Lena

(b) Pepper

(c) Boat

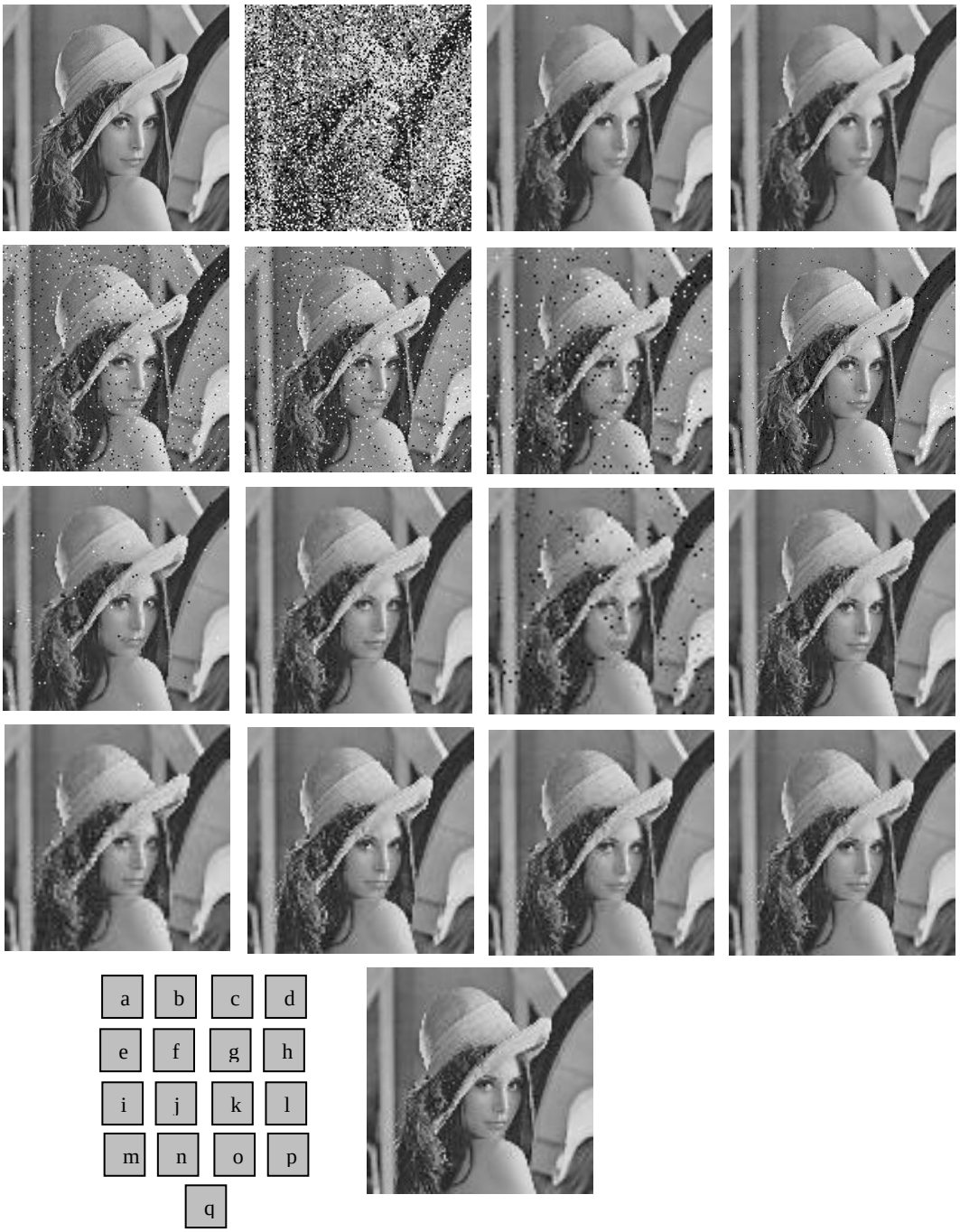


Fig.3.17 Performance of various filters for Lena image with noise density 40% (SPN)

(a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) CWM (f) TSM (g) AMF(h) PSM (i) SMF (j) BDND (k) ANDS (l) REIR (m) IDPSM (n) IRIC (o) ASF-I (p) ASF-II (q) IASF

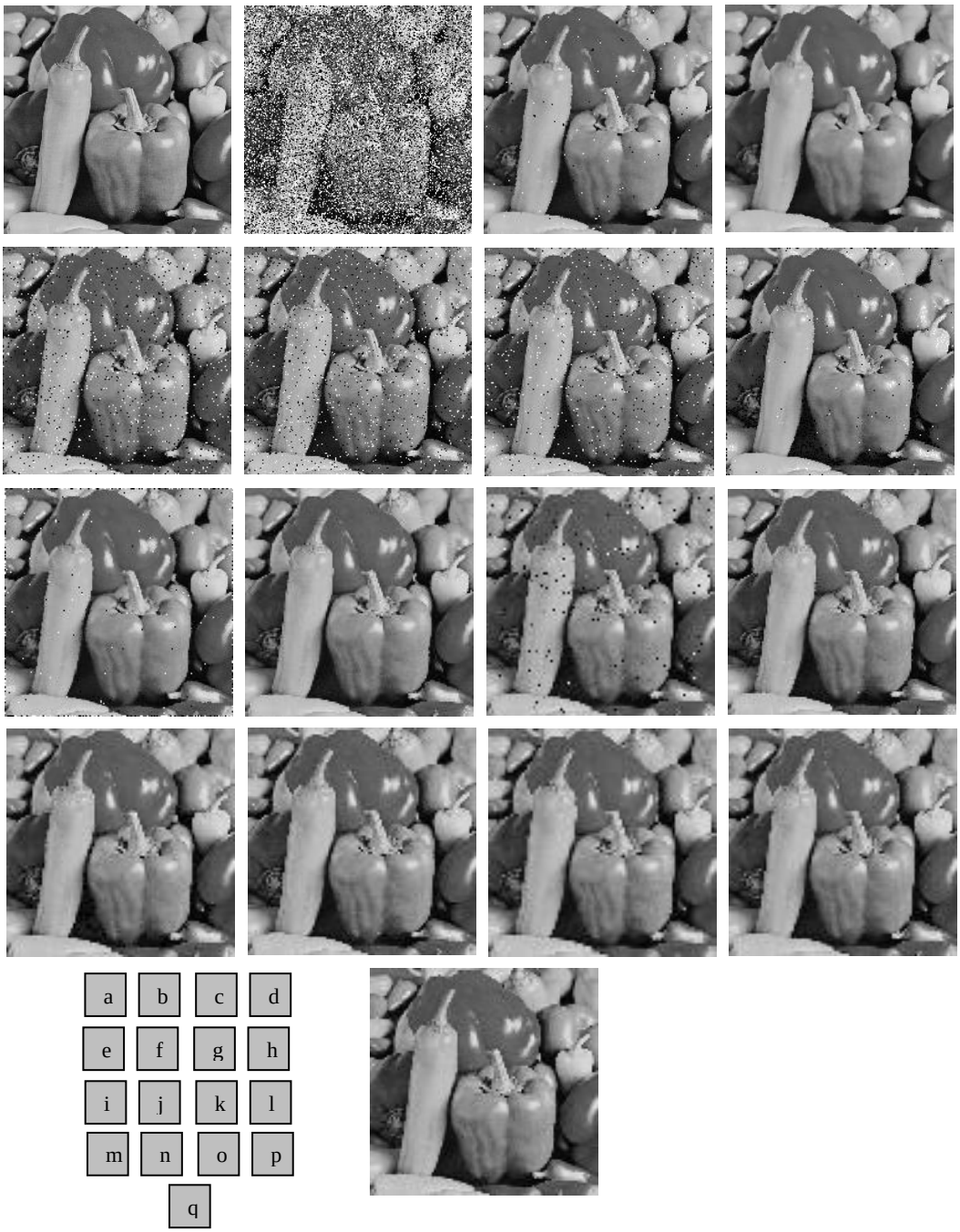


Fig. 3.18 Performance of various filters for Pepper image with noise density 40% (SPN)

(a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) CWM (f) TSM (g) AMF (h) PSM (i) SMF (j) BDND (k) ANDS (l) REIR (m) IDPSM (n) IRIC (o) ASF-I (p) ASF-II (q) IASF

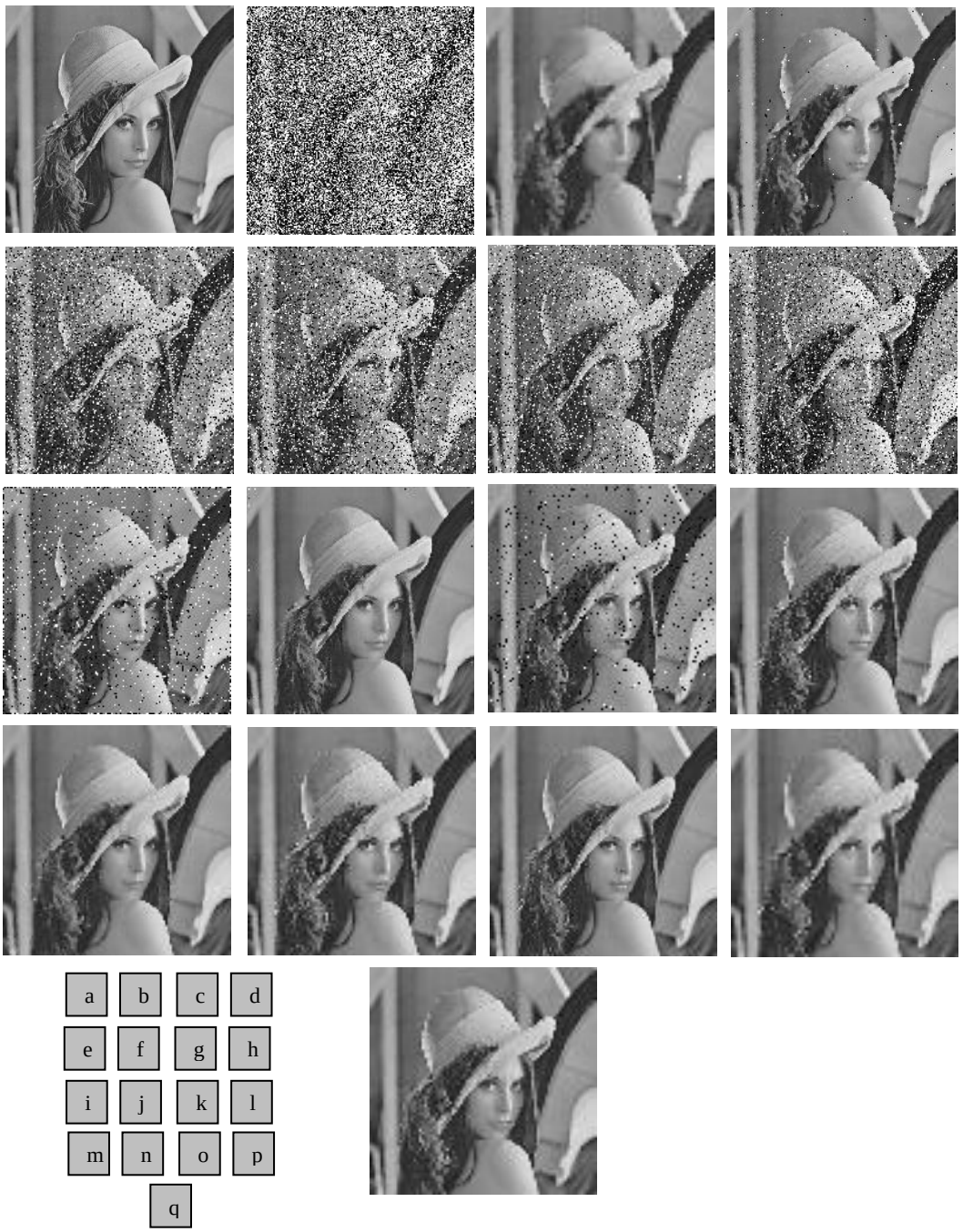


Fig. 3.19 Performance of various filters for Lena image with noise density 60% (SPN)

(a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) CWM (f) TSM (g) AMF(h) PSM (i) SMF (j) BDND (k) ANDS (l) REIR (m) IDPSM (n) IRIC (o) ASF-I (p) ASF-II (q) IASF

Fig. 3.20 Performance of various filters for Pepper image with noise density 60% (SPN)

(a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) CWM (f) TSM (g) AMF (h) PSM (i) SMF (j) BDND (k) ANDS (l) REIR (m) IDPSM (n) IRIC (o) ASF-I (p) ASF-II (q) IASF

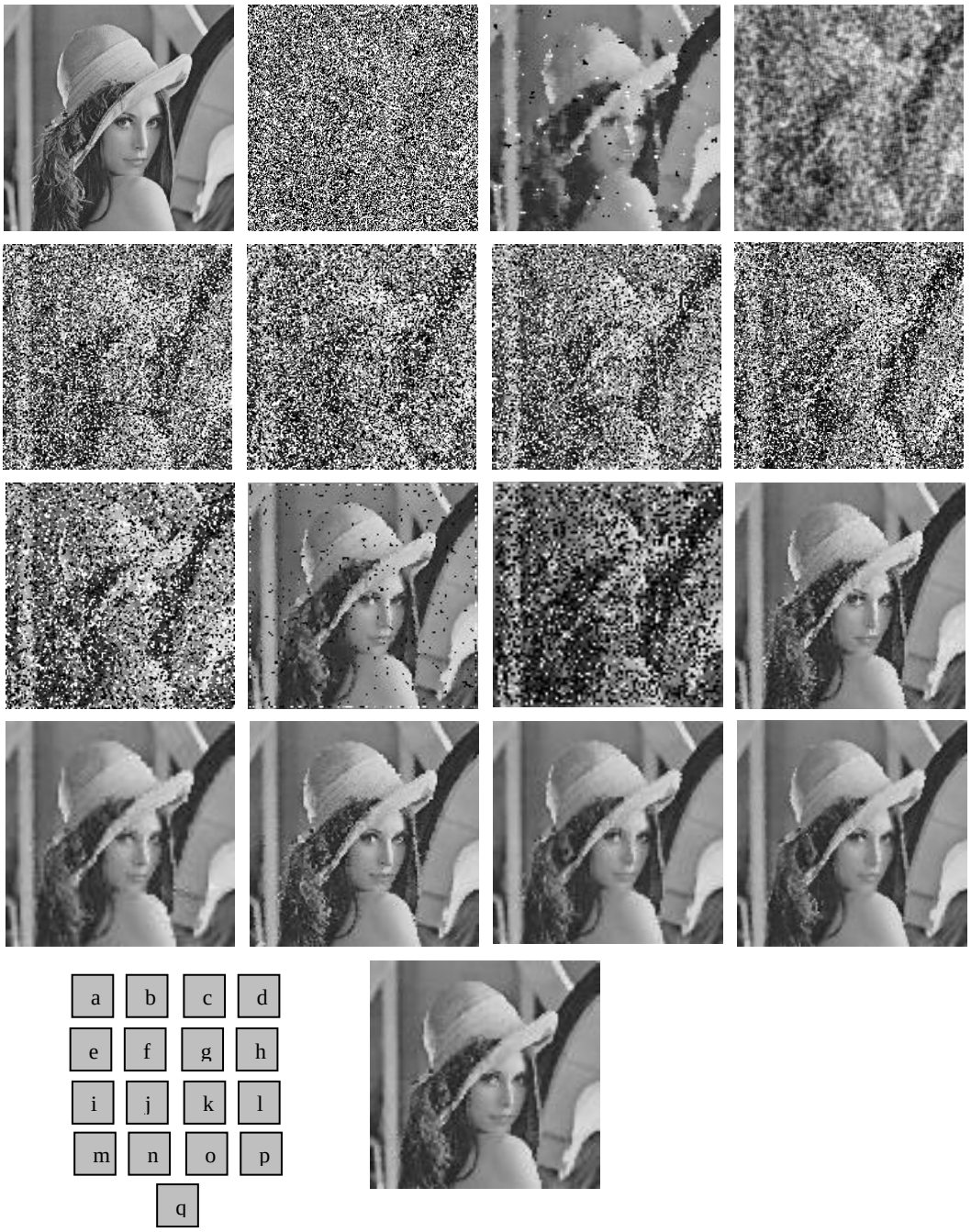


Fig. 3.21 Performance of various filters for Lena image with noise density 80% (SPN)

(a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) CWM (f) TSM (g) AMF(h) PSM (i) SMF (j) BDND (k) ANDS (l) REIR (m) IDPSM (n) IRIC (o) ASF-I (p) ASF-II (q) IASF

Fig. 3.22 Performance of various filters for Pepper image with noise density 80% (SPN) (a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) CWM (f) TSM (g) AMF(h) PSM (i) SMF (j) BDND (k) ANDS (l) REIR (m) IDPSM (n) IRIC (o) ASF-I (p) ASF-II (q) IASF

3.9. Conclusion

It is observed from simulation results that the proposed filters perform better than the existing methods for suppression of SPN. The proposed methods: ASF-I, ASF-II and IASF exhibit quite superior performance compared to other methods as they yield high PSNR, MSE, UQI and IEF. ASF-I shows its better performance in 40% noise density, ASF-II shows high performance up to 30% noise densities and whereas IASF perform better up 80% of noise densities. The performance of a filter depends on its ability to identify a noisy pixel and replace it with an efficient estimation. The IASF algorithm is iterative in nature which makes it more efficient in proper noise detection. Further, in both the algorithms, adaptive filtering window helps to retain the edges and fine details of an image. Hence, these two filters show better noise suppressing capability without yielding any appreciable distortion and blur. ANDS and ASF-II shows their best performance only under low density (10%).

It is also observed that the ASF-I, ASF-II and IASF preserve the edges and fine details of an image very well, as observed from Fig. 3.17-3.22, compared to other filters.

Fig. 3.13 shows the graphical representation of PSNR values. The filters ASF-I, ASF-II and IASF show the better performance.

In one or two occasions the IEF value of BDND filter shows good results (for Boat and Pepper images) for medium noise density, but it fails to perform well under high noise density.

For online and real-time applications the system must have small execute time T_E with less complexity. Table-3.15 indicates that the filters ASF-I and ASF-II show its best performance along with good filtering operation.

They are having the following advantages:

i. Less computational complexity compared to any other methods
ii. The noise suppressing capacity is good in all types of test images.
iii. They retain the detailed information very well as compared to other filters.

Thus, the proposed filters: **ASF-I, ASF-II** and **IASF** are observed to be very good special-domain image denoising filters for efficient suppression of salt-and-pepper impulse noise.

Chapter 4

Development of Novel Filters for Suppression of Random-Valued Impulse Noise

4

Preview

Filtering a random-valued impulse noise (RVIN) is also accomplished in two stages: detection of noisy pixels and replacement of these pixels with the help of an estimator. The difference in gray level between a noisy pixel and a noise-free neighbor is not always appreciable when an image is corrupted with RVIN. Filtering a random-valued impulse noise is far more difficult than filtering a fixed-valued impulse noise. With the basic Classifier-Filter proposition (depicted in Section-3.1) and employing median estimator for filtration, some novel algorithms are developed to suppress RVIN of low to medium densities quite efficiently.

In the next section, two important statistical parameters: median of the absolute deviations from the median (MAD) and pixel-wise MAD (PWMAD) are described. Some novel filters are developed, in section-4.2 and 4.3, based on MAD and PWMAD employing the basic BCF and ICF-II classifier-filter structures.

The following topics are covered in this chapter.

- MAD and PWMAD
- Adaptive Window based Pixel Wise MAD (AW-PWMAD) Algorithm
- Adaptive Local Thresholding with MAD (ALT-MAD) Algorithm
- Simulation Results
- Conclusion

4.1 MAD and PWMAD

In this section, some statistical parameters that are important and useful in image processing are described. These parameters are employed to develop some novel image denoising algorithms to suppress RVIN as well as SPN very efficiently.

Median of Absolute Deviations from the Median (MAD) and Pixel-Wise MAD (PWMAD):

A robust statistical estimation parameter, median of absolute deviation from median (MAD) [112], is defined by:

$$MAD_{k,l}(i,j) \equiv med\left(\left\{\left|g_{k,l}(i,j) - m\right|\right\}\right) \tag{4.1}$$

where, $m = med(\{g(i,j)\})$

MAD considers deviation from the median of whole image, i.e., it takes a global statistical parameter that may or may not represent a truth in a local framework. To overcome this limitation, a new statistical parameter: pixel-wise MAD (PWMAD) is defined that considers deviation from median of local samples. This is expected to be a robust estimator of a random variable.

PWMAD [122], is defined by:

$$PWMAD_{k,l}(i,j) \equiv med\left(\left\{\left|g_{k,l}(i,j) - m_{k,l}(i,j)\right|\right\}\right) \tag{4.2}$$

where, $m_{k,l}(i,j) = m(i,j) = med\left(\left\{g_{k,l}(i,j)\right\}\right), -r < k,l < +r$

For simplicity, new symbols $\mathbb{U}$ and $\mathbb{V}$ are introduced to represent MAD and PWMAD respectively, i.e.,

$$\mathbb{U}(i,j) = MAD_{k,l}(i,j) \tag{4.3}$$

$$\mathbb{V}(i,j) = PWMAD_{k,l}(i,j) \tag{4.4}$$

4.2 Adaptive Window based Pixel-Wise MAD (AW-PWMAD) Algorithm [P8]

Under the Classifier-Filter paradigm, an iterative classifier-filter: ICF-2 paradigm was introduced in Section 3.1. A novel adaptive-window filtering scheme is developed under this ICF-2 framework, shown in Fig.3.1(d), that employs fixed window for decision making.

4.2.1 Noise Detection Algorithm

The decision is based on robust estimators like MAD, PWMAD and their difference. A modified MAD is computed, under iterative framework, given by:

$$\mathbb{U}_{k,l}^{(n+1)}(i,j) = \left|\mathbb{U}_{k,l}^{(n)}(i,j) - \mathbb{V}_{k,l}^{n}(i,j)\right| \tag{4.5}$$

n = 0,1,2,3,.....,N-1

After N iterations, the modified MAD, $\mathbb{U}^N$, is expected to contain noise only. To classify the input data as corrupted (noisy) or uncorrupted (noise-free), the following hypothesis (noise-detection algorithm) is made to generate a binary flag image, $b(i,j)$.

IF $\mathbb{U}^N(i,j) \geq T$,

THEN $b(i,j) = 1$

ELSE $b(i,j) = 0$

where T is a threshold whose optimum value is evaluated by searching for best performance in terms of PSNR in a separate experiment, discussed in Section 4.2.3.

Fig. 4.1 shows the flowchart for this noise detection algorithm.

4.2.2 Estimation Algorithm

Fig. 4.2 shows the flowchart for estimation algorithm. The binary image $b(i,j)$ controls the filtering operation. Based on binary flag, no filtering is applied to the uncorrupted pixels (i.e., $b(i,j) = 0$), while the switching median filter with an adaptively determined window size is applied to each corrupted pixel (i.e., $b(i,j) = 1$).

Starting with (3×3) filtering window iteratively extends outward by one pixel in all the four sides of the window, provided that the number of uncorrupted pixels, $C_{w2,}$ is less than *half* of the total number of pixels within the filtering window. The maximum filtering window size is limited to (7×7) to avoid undesired distortion and blurring. Since the central pixel has been detected as noisy, it will not participate in the filtering process. Only the pixels, which are classified as noise-free in filtering window, will participate in median filtering process. This will, in turn, yield a better filtering result with less distortion.

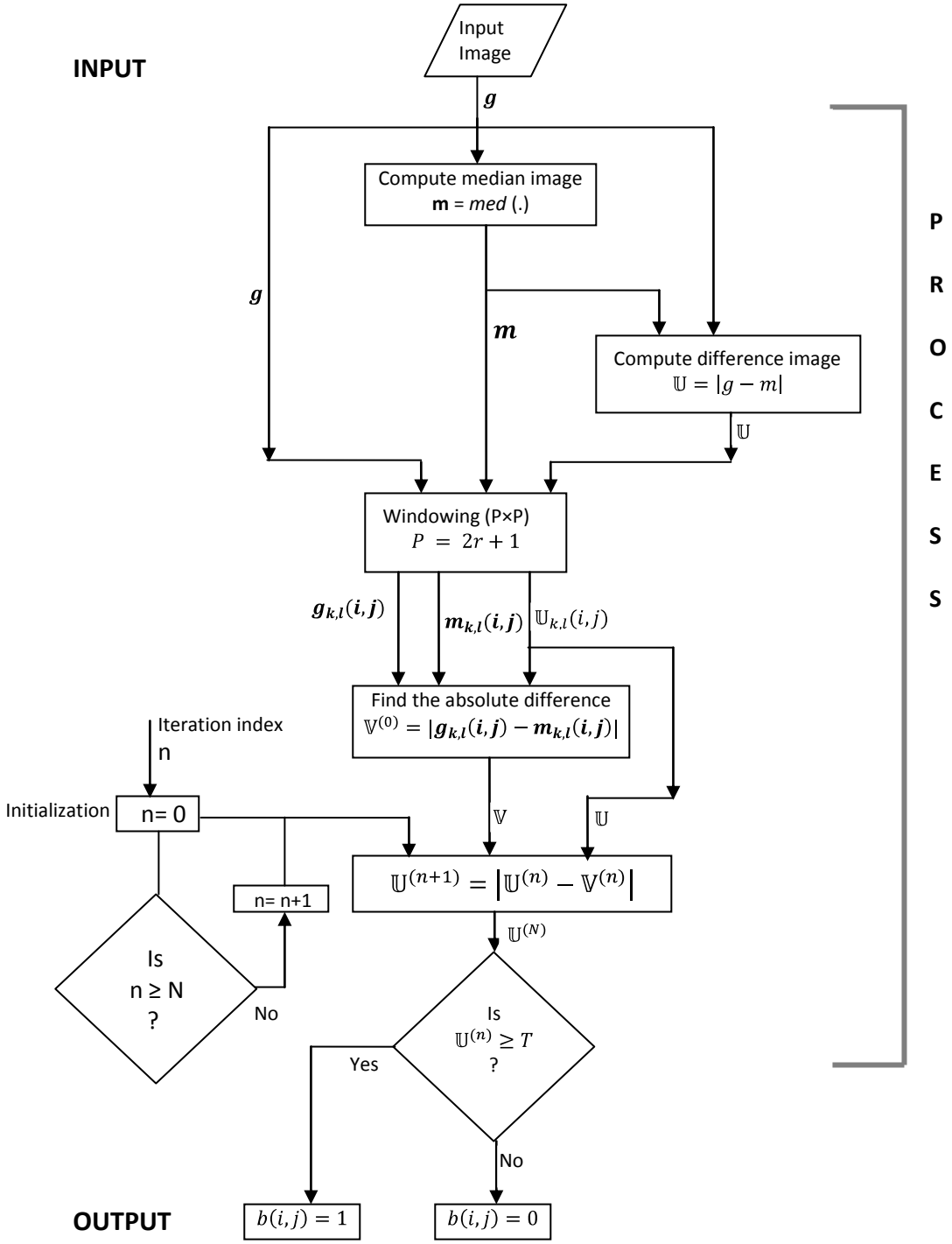


Fig. 4.1 Flowchart of noise detection

4.2.3 Optimizing the Threshold

In order to optimize the value of threshold, various simulation experiments are conducted on standard test images, corrupted with RVIN of different noise densities. The performance is evaluated in terms of PSNR. The simulated results of *Lena* test image is tabulated in table Table-4.1. It is observed that the proposed system yields high performance, in terms of PSNR, for the threshold, T ∈ [2, 5].Thus, an optimized value of threshold, T, i.e., $T_{optimal}$ is taken as **3**.

Table-4.1: Performance of AW-PWMAD filter in terms of PSNR for different Threshold T, operated on Lena image corrupted with RVIN under varies noise densities

Sl. No	Threshold T	RVIN Noise (in%)		
		5	**10**	**15**
1	1	41.01	33.45	28.03
2	2	41.06	33.32	28.04
3	**3**	41.06	34.23	**28.19**
4	4	41.01	33.72	**28.19**
5	**5**	**41.22**	**34.57**	27.93
6	10	41.67	34.17	27.98
7	15	41.93	34.22	27.98
8	20	41.90	33.01	27.75
9	25	41.54	33.06	27.48
10	30	40.48	33.66	27.70
11	35	40.68	33.41	27.52
12	40	40.22	33.80	27.52
13	50	36.07	31.20	26.08

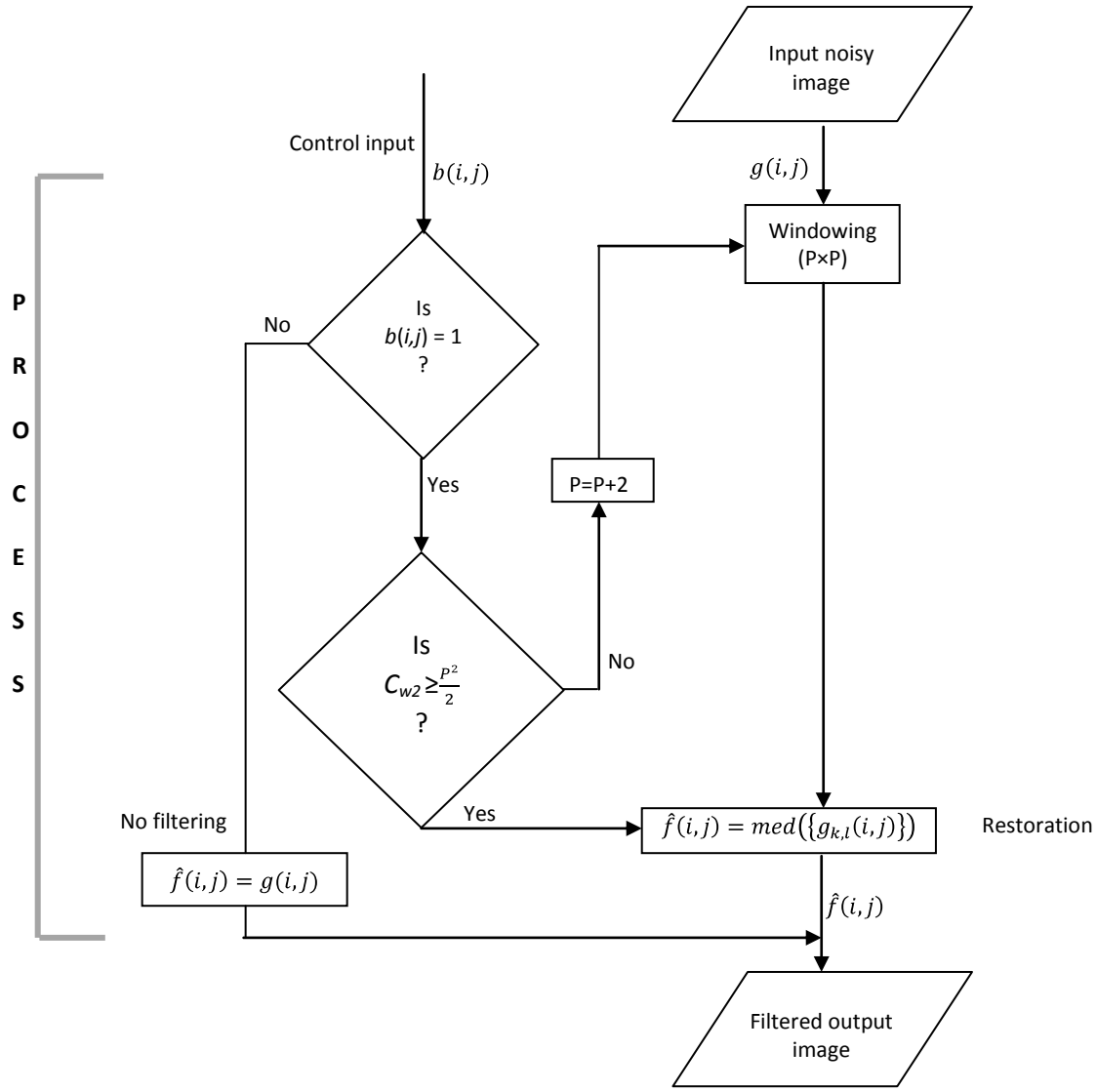


Fig. 4.2 Flowchart for estimation algorithm

4.3 Adaptive Local Thresholding with MAD (ALT-MAD) Algorithm [P9]

Another proposed method for denoising the random-valued and fixed-valued impulse noise employs BCF framework shown in Fig. 3.1(b) in Section 3.1. A modified MAD based algorithm along with a local adaptive threshold is exploited for pixel classification. The noisy pixel is replaced with median of uncorrupted pixels in the filtering window of adaptively varied size.

The proposed method is a modified version of the AW-PWMAD algorithm described in Section-4.2. The modifications are: (i) No iteration is used for noise detection so as to reduce computational complexity, and (ii) Use of adaptive local threshold for better classification of pixels.

The selection of threshold *T* is important in pixel-classification. If the value of *T* is set too high, it omits certain portion of noisy pixels from the noise map. On the other hand, if *T* is set too low, image details will be treated as noise, and the overall image quality will be degraded. To overcome this problem a locally adaptive threshold, based on MAD value of the window, is proposed. Hence, the performance of the proposed method is better than the previous method i.e., AW-PWMAD.

Three threshold functions are suggested and employed in this algorithm. Thus, three different versions, namely, ALT-MAD-1, ALT-MAD-2 and ALT-MAD-3 are developed. They are observed to be quite efficient in noise detection and filtering.

Proposed functions:

ALT-MAD-1

$$T = \begin{cases} a, & MAD(i,j) < b \\ a + \dfrac{b-a}{a} \times MAD(i,j), & MAD(i,j) \geq b \end{cases} \tag{4.6}$$

ALT-MAD-2

$$T = \begin{cases} a, & MAD(i,j) < b \\ a + \dfrac{b-a}{a} \times MAD(i,j) \times (.1)^{\lambda}, & MAD(i,j) \geq b \end{cases} \quad where\ \lambda > 1 \tag{4.7}$$

ALT-MAD-3

$$T = \begin{cases} a, & MAD(i,j) < b \\ a + \dfrac{b-a}{a} \times MAD(i,j) \times (.1)^{\lambda}, & MAD(i,j) \geq b \end{cases} \quad where\ \lambda < 1 \tag{4.8}$$

4.3.1 Optimizing Parameters

Simulation experiments are conducted on standard test images corrupted with RVIN of different noise densities to find optimal values for the parameters: a, b and λ. The performance is evaluated in terms of PSNR. The simulated results for *Lena* test image are tabulated in Table-4.2, 4.3 and 4.4. It is observed that the proposed filter ALT-MAD-1 yields high performance, in terms of PSNR, for the parameter-values:

a=10 and b=40, for ALT-MAD-2, a=10, b=40 and λ = 3.3 and for ALT-MAD-3, a=10, b=40 and λ =0.3.

Table-4.2: Performance of ALT-MAD-1 filter in terms of PSNR for different a and b values, operated on Lena image corrupted with RVIN under various noise densities

Sl. No	a	b	RVIN Noise (%)		
			10	**20**	**30**
1	5	30	35.21	23.72	17.24
2	10		35.57	23.81	17.56
3	20		35.41	23.56	17.33
4	30		35.33	23.11	17.41
5	5	**40**	35.30	24.00	17.80
6	**10**		**36.22**	**24.76**	**18.40**
7	20		34.84	23.71	17.75
8	30		34.51	23.01	17.26
9	5	50	36.01	23.74	17.76
10	10		**36.55**	24.10	17.96
11	20		35.71	24.07	18.00
13	30		34.65	24.00	17.57
14	5	60	32.48	22.41	16.86
15	10		33.05	22.63	16.99
16	20		33.71	23.00	17.09
17	30		33.75	23.18	17.03

Table-4.3: Performance of ALT-MAD-2 filter in terms of PSNR for different λ, a=10 and b=40 values, operated on Lena image corrupted with RVIN under various noise densities

Sl. No	% of Noise	λ								
		2	3	4	5	3.1	3.2	**3.3**	3.4	3.5
1	10	35.78	36.01	36.11	35.57	36.01	36.11	**35.90**	35.31	35.11
2	20	24.00	24.40	23.91	23.47	24.86	24.84	**25.02**	24.71	24.70
3	30	17.87	18.21	17.79	17.22	18.49	18.55	**18.78**	18.72	18.41

Table-4.4: Performance of ALT-MAD-3 filter in terms of PSNR for different λ, a=10 and b=40 values, operated on Lena image corrupted with RVIN under various noise densities

Sl. No	% of Noise	λ							
		0.1	0.2	**0.3**	0.4	0.5	0.6	0.7	0.8
1	10	36.26	37.23	**37.39**	36.92	36.21	35.11	34.51	34.02
2	20	25.44	25.13	**25.33**	25.00	24.72	24.23	23.72	22.89
3	30	19.29	19.18	**19.05**	19.11	18.71	18.13	17.78	17.44

Extensive simulations are conducted on different gray scale test images and results are presented in Section-4.4.

4.4 Simulation Results

All the simulation experiments are carried out on a MATLAB-7.4 platform that sits over a Windows-XP operating system.

The performances of proposed and existing filters are tested on different test images. The test images employed for simulations are: *Lena, Boat, and Pepper*. All of them are 8-bit gray scale images of size 512×512.

Image metrics: PSNR, MSE, UQI and IEF are evaluated for performance-evaluation of filters.

The PSNR values of different filters are tabulated in the tables: Table-4.5 through Table-4.7. The MSE values are presented in tables: Table-4.8 through Table-4.10 whereas UQI results are presented in tables: Table-4.11 through Table-4.13. Further, the tables: Table-4.14 through Table-4.16 demonstrate the performance of filters in terms of IEF. The best results are highlighted in bold font for quick analysis in the tables.

The proposed filters works better even for the images corrupted by salt and pepper noise. The Table 4.17 shows the performance of filters both in salt-and-pepper and random-valued impulse noise. The PSNR value is used as a performance measuring metrics. The best results are highlighted for quick analysis.

The graphical representation of PSNR, MSE, UQI, and IEF of proposed filters and some existing filters are illustrated in figures: Fig.4.3 through Fig. 4.6 for easy analysis. For subjective evaluation, the images are corrupted with noise density 10% and 20% are applied to different filters and the resulted output images are shown in figures: Fig. 4.7 through Fig. 4.10. The test images: Lena and Pepper are used for subjective evaluation.

Conclusions are drawn in the next section.

Table-4.5: Filtering performance of various filters in terms of PSNR (dB) for RVIN
Test Image: *Lena*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	33.95	29.86	24.56	20.57	17.31	14.85
2	ATM[3×3]	33.66	30.02	25.81	22.22	18.86	16.15
3	TSM	34.86	30.10	24.23	21.22	17.63	15.10
4	PSM	34.93	31.55	28.13	24.19	20.89	17.36
5	PWMAD	33.52	28.28	22.28	19.01	15.84	15.84
6	AMAD	36.01	31.94	27.01	24.22	21.05	17.56
7	AW-PWMAD[P8]	41.56	33.54	27.84	23.37	20.07	17.02
8	ALT-MAD-1	48.52	36.88	30.77	25.01	21.21	18.54
9	ALT-MAD-2	48.18	36.21	30.21	24.71	20.98	18.13
10	**ALT-MAD-3**	**49.42**	**37.02**	**30.89**	**25.22**	**21.78**	**18.98**

Table-4.6: Filtering performance of various filters in terms of PSNR (dB) for RVIN
Test Image: *Pepper*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	34.22	29.88	24.66	20.37	17.18	14.72
2	ATM[3×3]	33.11	29.76	26.04	22.32	19.09	16.39
3	TSM	36.73	30.98	26.65	22.46	18.68	15.83
4	PSM	37.75	32.53	28.17	23.19	18.88	17.45
5	PWMAD	36.12	29.85	23.97	19.95	16.63	14.11
6	AMAD	39.02	32.94	28.01	23.22	19.05	18.06
7	AW-PWMAD[P8]	42.08	33.69	28.17	23.43	19.49	17.22
8	ALT-MAD-1	44.65	35.08	28.24	23.15	19.69	16.78
9	ALT-MAD-2	**49.24**	**35.78**	28.07	23.32	19.40	16.73
10	ALT-MAD-3	47.72	35.76	**28.49**	**23.63**	**19.72**	**18.24**

Table-4.7: Filtering performance of various filters in terms of PSNR (dB) for RVIN
Test Image: *Boat*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	30.48	27.75	23.37	19.68	16.75	14.35
2	ATM[3×3]	29.81	27.49	24.49	21.08	18.23	15.73
3	TSM	32.15	29.46	25.25	21.48	17.19	15.23
4	PSM	32.35	30.17	26.97	22.85	18.65	16.79
5	PWMAD	31.78	27.63	23.40	19.26	16.13	13.65
6	AMAD	34.37	31.16	27.04	23.01	19.12	16.38
7	AW-PWMAD[P8]	38.86	32.42	27.34	23.06	19.49	16.66
8	ALT-MAD-1	46.31	**34.92**	27.77	22.77	19.49	16.76
9	ALT-MAD-2	**48.61**	34.91	27.33	23.00	19.37	16.63
10	ALT-MAD-3	47.91	35.35	**28.31**	**23.24**	**19.92**	**17.00**

Table-4.8: Filtering performance of various filters in terms of MSE for RVIN
Test Image: *Lena*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	20.75	66.57	227	569.3	1207	2125
2	ATM[3×3]	27.97	64.62	170.4	389.1	825.5	1524
3	TSM	19.86	48.2	159.8	392.4	891.4	1709
4	PSM	10.46	28.66	79.41	260	495	948.4
5	PWMAD	18.93	72.19	254.1	687	1420	2515
6	AMAD	7.91	24.59	73.53	254	501	958
7	AW-PWMAD[P8]	4.52	28.73	106.8	299	639	1288
8	ALT-MAD-1	1.06	15.77	67.59	215.1	503.76	939.23
9	ALT-MAD-2	1.15	13.72	66.68	227.36	520.32	999.78
10	ALT-MAD-3	**0.98**	**13.47**	**66.43**	**211.67**	**486.2**	**898.45**

Table-4.9: Filtering performance of various filters in terms of MSE for RVIN
Test Image: *Pepper*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	24.55	66.74	222.2	597	1244	2189
2	ATM[3×3]	31.73	68.6	161.7	380.9	800	1491
3	TSM	13.81	51.88	140.5	368.5	880	1698
4	PSM	10.89	36.25	99.04	315.5	802	1250
5	PWMAD	15.89	67.19	260.4	656.9	1411	2518
6	AMAD	8.15	33	102.7	295.6	680.3	1251
7	AW-PWMAD[P8]	4.02	27.77	98.91	291.7	751.1	1231
8	ALT-MAD-1	2.24	19.19	98.06	314.40	708.22	1381
9	ALT-MAD-2	**0.98**	**17.24**	95.43	302.64	745.38	1380
10	ALT-MAD-3	1.09	18.05	**93.26**	**281.5**	**697.31**	**1211**

Table-4.10: Filtering performance of various filters in terms of MSE for RVIN
Test Image: *Boat*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	58.15	111.7	298.7	672.8	1371	2383
2	ATM[3×3]	67.89	115.8	230.7	506.5	975.3	1735
3	TSM	39.56	73.48	193.9	462.3	1050	1948
4	PSM	37.80	62.44	130.6	367.6	764	1351
5	PWMAD	15.88	67.19	260.4	656	1412	2519
6	AMAD	23.77	49.75	128.5	282.8	776	944
7	AW-PWMAD[P8]	8.44	37.24	120	324	730	1403
8	ALT-MAD-1	1.69	20.89	109.01	343.44	731.12	1377
9	ALT-MAD-2	**1.01**	21.03	115.43	321.1	745	1380
10	ALT-MAD-3	1.25	**17.23**	**92.23**	**320.32**	**717**	**1322**

Table-4.11: Filtering performance of various filters in terms of UQI for RVIN
Test Image: *Lena*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	0.995	0.985	0.953	0.890	0.791	0.679
2	ATM[3×3]	0.993	0.986	0.965	0.925	0.855	0.762
3	TSM	0.996	0.988	0.967	0.921	0.836	0.725
4	PSM	0.997	0.993	0.983	0.952	0.886	0.810
5	PWMAD	0.995	0.984	0.947	0.869	0.760	0.635
6	AMAD	0.998	0.994	0.984	0.953	0.8927	0.829
7	AW-PWMAD[P8]	0.9994	0.993	0.977	0.938	0.877	0.776
8	ALT-MAD-1	0.9998	0.9970	0.9858	0.9552	0.9006	0.8278
9	ALT-MAD-2	0.9997	0.9966	0.9854	0.9545	0.8974	0.8173
10	**ALT-MAD-3**	**0.9998**	**0.9970**	**0.9896**	**0.9578**	**0.9039**	**0.8344**

Table-4.12: Filtering performance of various filters in terms of UQI for RVIN
Test Image: *Pepper*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	0.995	0.988	0.963	0.907	0.821	0.718
2	ATM[3×3]	0.994	0.988	0.973	0.941	0.884	0.787
3	TSM	0.997	0.991	0.976	0.940	0.867	0.769
4	PSM	0.998	0.993	0.983	0.946	0.833	0.801
5	PWMAD	0.997	0.988	0.956	0.897	0.799	0.683
6	AMAD	0.998	0.994	0.982	0.949	0.888	0.761
7	AW-PWMAD[P8]	0.9995	0.995	0.983	0.943	0.887	0.801
8	ALT-MAD-1	0.9996	0.9965	0.9834	0.9485	0.8805	0.8043
9	ALT-MAD-2	**0.9998**	**0.9970**	0.9823	0.9504	0.8850	0.8037
10	ALT-MAD-3	0.9998	0.9965	**0.9842**	**0.9536**	**0.8928**	**0.8150**

Table-4.13: Filtering performance of various filters in terms of UQI for RVIN
Test Image: *Boat*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	0.986	0.974	0.935	0.865	0.757	0.639
2	ATM[3×3]	0.984	0.974	0.950	0.899	0.825	0.726
3	TSM	0.990	0.983	0.957	0.903	0.804	0.684
4	PSM	0.991	0.985	0.971	0.943	0.857	0.744
5	PWMAD	0.990	0.974	0.935	0.849	0.730	0.597
6	AMAD	0.994	0.988	0.971	0.938	0.849	0.717
7	AW-PWMAD[P8]	0.998	0.991	0.973	0.931	0.840	0.751
8	ALT-MAD-1	0.9996	0.9956	0.9756	0.9267	0.8558	0.7574
9	ALT-MAD-2	**0.9998**	0.9951	0.9745	0.9304	0.8524	0.7523
10	ALT-MAD-3	0.9997	**0.9960**	**0.9782**	**0.9389**	**0.8582**	**0.7649**

Table-4.14: Filtering performance of various filters in terms of IEF for RVIN
Test Image: *Lena*

Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	42.08	26.38	11.78	6.13	3.63	2.46
2	ATM[3×3]	31.49	24.26	15.44	8.98	5.29	3.47
3	TSM	51.40	34.79	17.10	9.01	4.92	3.07
4	PSM	83.45	61.03	26.13	**12.59**	**7.49**	4.08
5	PWMAD	46.26	24.35	10.43	5.11	3.07	2.09
6	AMAD	78.05	48.48	24.67	10.86	6.14	**5.96**
7	AW-PWMAD[P8]	191.8	62.10	24.62	11.77	6.82	4.09
8	ALT-MAD-1	1020	87.73	26.56	11.21	5.99	3.87
9	ALT-MAD-2	604	77.46	25.51	11.15	5.80	3.64
10	ALT-MAD-3	**1020**	**87.83**	**27.50**	11.43	6.21	4.10

Table-4.15: Filtering performance of various filters in terms of IEF for RVIN
Test Image: *Pepper*

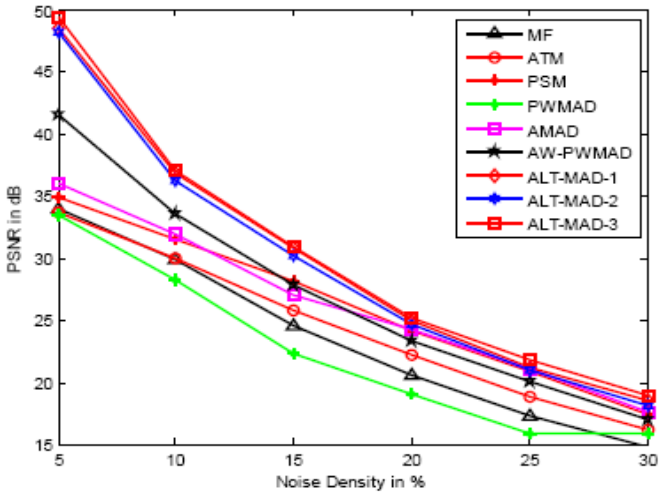
Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	35.35	25.86	11.63	5.80	3.48	2.37
2	ATM[3×3]	27.13	25.15	15.81	8.96	5.35	3.47
3	TSM	61.98	33.69	18.28	9.32	4.89	3.04
4	PSM	78.41	47.15	26.20	10.68	7.09	4.00
5	PWMAD	53.70	25.72	9.94	5.19	3.03	2.04
6	AMAD	108.8	52.21	25.29	**14.00**	**8.42**	**5.69**
7	AW-PWMAD[P8]	215.7	62.00	26.03	12.20	6.55	4.17
8	ALT-MAD-1	380.29	85.86	26.45	10.97	6.04	3.74
9	ALT-MAD-2	**1300**	**99.33**	**28.03**	11.40	5.82	3.75
10	ALT-MAD-3	781	85.31	27.63	12.81	6.21	4.01

Table-4.16: Filtering performance of various filters in terms of IEF for RVIN
Test Image: *Boat*

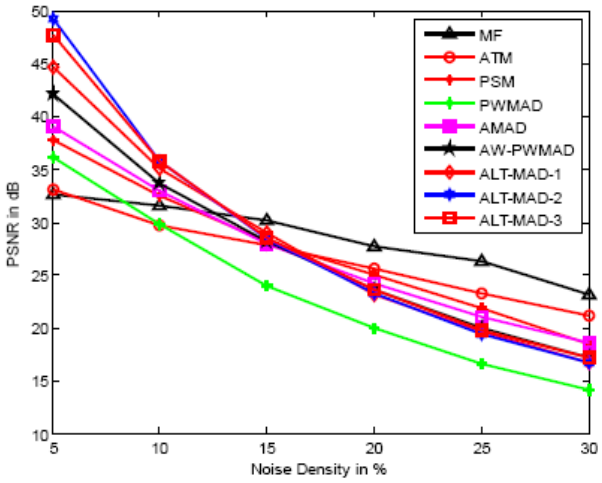
Sl. No	Filters	% of Noise (Random-valued Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
1	MF [3×3]	16.24	16.87	9.51	5.61	3.46	2.37
2	ATM[3×3]	14.18	16.40	12.27	7.50	4.85	3.27
3	TSM	23.55	25.79	14.69	8.12	4.49	2.89
4	PSM	24.77	30.40	21.72	10.13	6.41	4.14
5	PWMAD	22.38	16.85	9.62	4.89	2.98	2.02
6	AMAD	40.01	38.05	21.84	10.30	**8.20**	**5.99**
7	AW-PWMAD[P8]	111.0	50.35	23.68	**11.68**	6.50	4.03
8	ALT-MAD-1	615.58	89.68	26.19	11.03	6.41	4.09
9	ALT-MAD-2	**1052**	89.53	**29.38**	11.61	6.28	4.01
10	ALT-MAD-3	890.26	**98.88**	24.74	11.39	6.58	4.29

Table-4.17: Comparison of Filtering performance of various filters in terms of PSNR (dB) for both SPN and RVIN. Test image: *Lena*

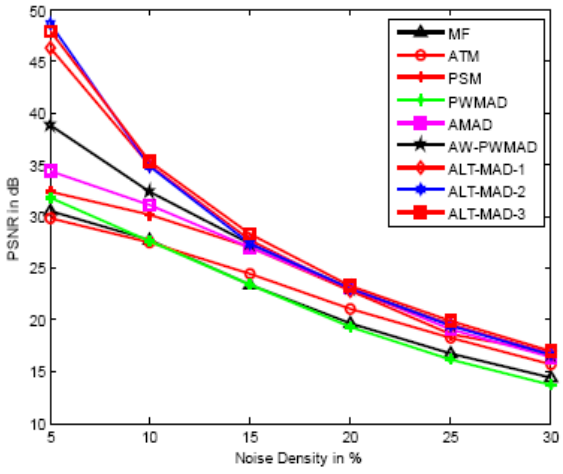
Sl. No	Filters	% Noise (SPN)			% Noise (RVIN)		
		10	**20**	**30**	**10**	**20**	**30**
1	MF [3×3]	33.74	27.28	21.75	29.86	20.57	14.85
4	ATM[3×3]	31.99	27.72	23.06	30.02	25.57	21.16
7	TSM	35.49	29.31	23.48	30.10	21.22	15.10
8	PSM	37.27	32.71	29.21	31.55	24.19	17.36
9	PWMAD	32.11	23.56	17.91	28.28	19.01	15.84
10	AMAD	37.13	30.32	24.27	31.94	24.22	17.56
11	AW-PWMAD	38.77	30.77	24.28	33.54	23.37	17.02
12	ALT-MAD-1	43.48	30.89	22.14	36.88	25.01	18.54
13	ALT-MAD-2	43.21	30.44	21.63	36.21	24.71	18.13
14	**ALT-MAD-3**	**44.50**	**32.75**	**25.11**	**37.02**	**25.22**	**18.98**



(a)



(b)



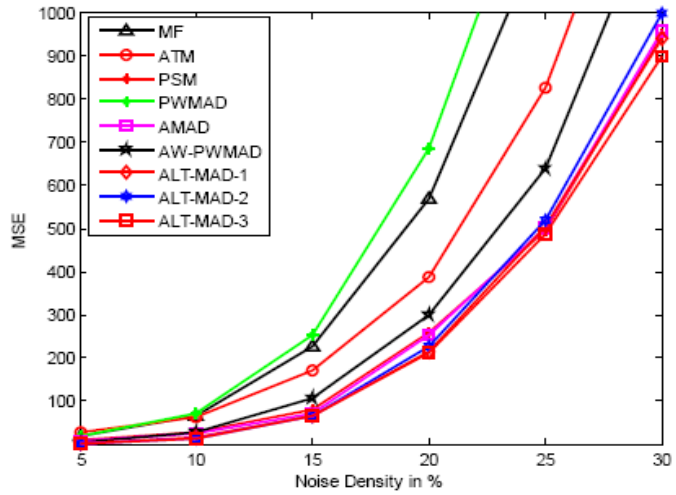
(c)

Fig. 4.3 Performance comparison of various filters in terms of PSNR (dB) for RVIN at different noise densities on the images:

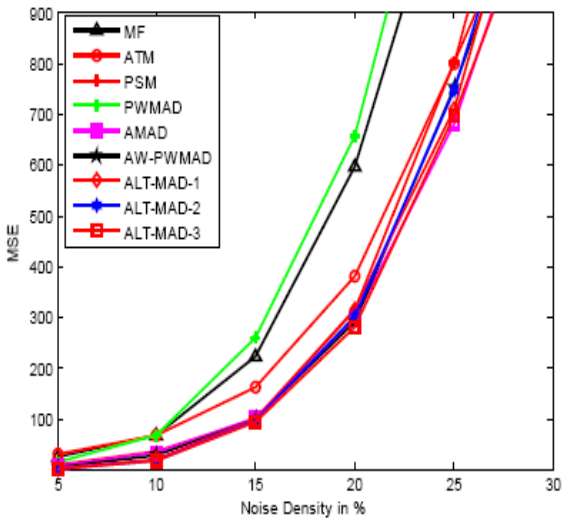
(a) Lena

(b) Pepper

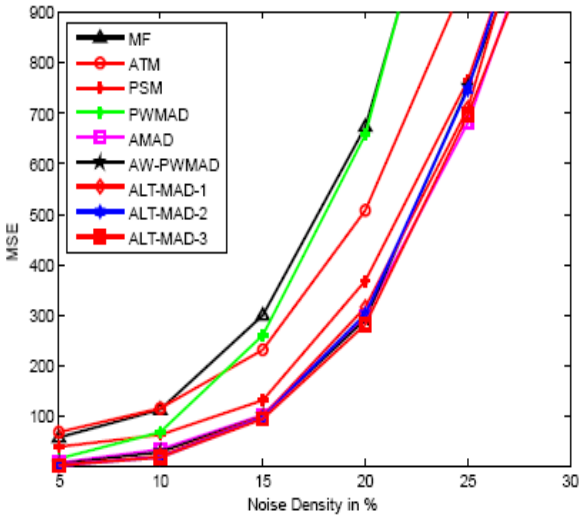
(c) Boat



(a)



(b)



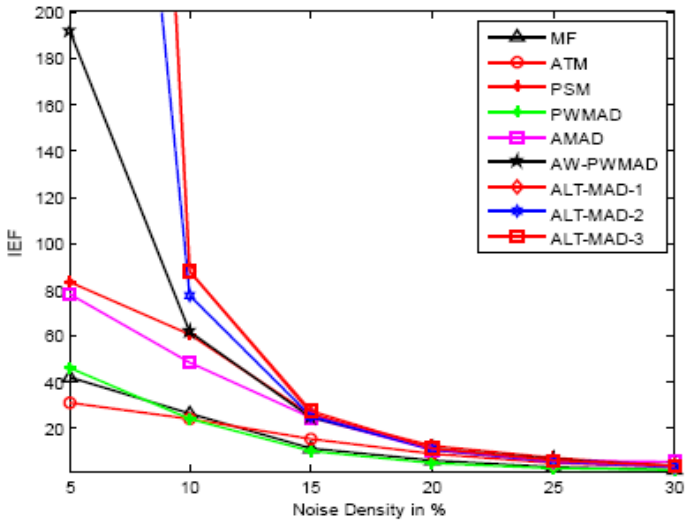
(c)

Fig. 4.4 Performance comparison of various filters in terms of MSE for RVIN at different noise densities on the images:

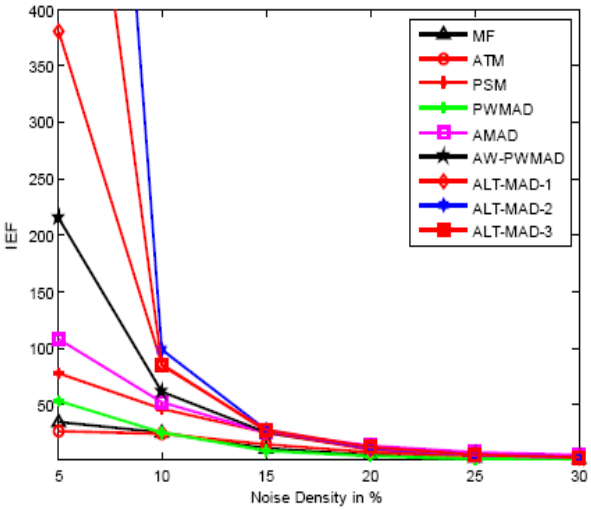
(a) Lena

(b) Pepper

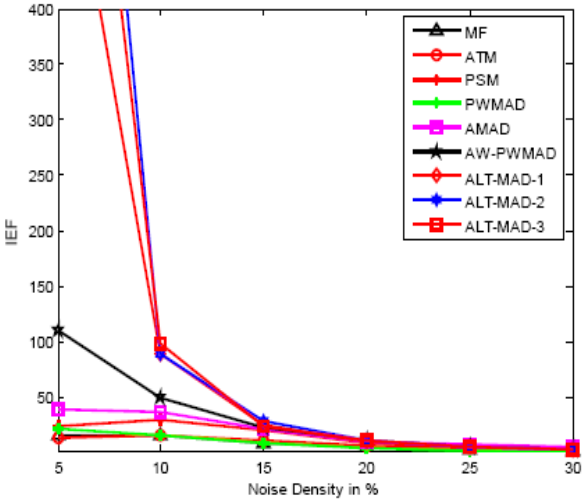
(c) Boat



(a)



(b)



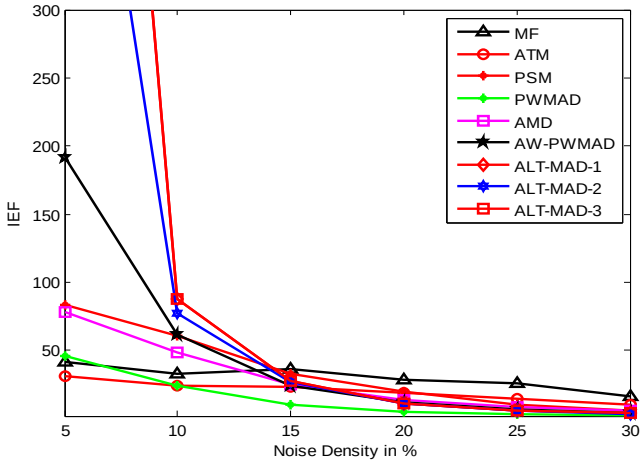
(c)

Fig. 4.5 Performance comparison of various filters in terms of UQI for RVIN at different noise densities on the images:

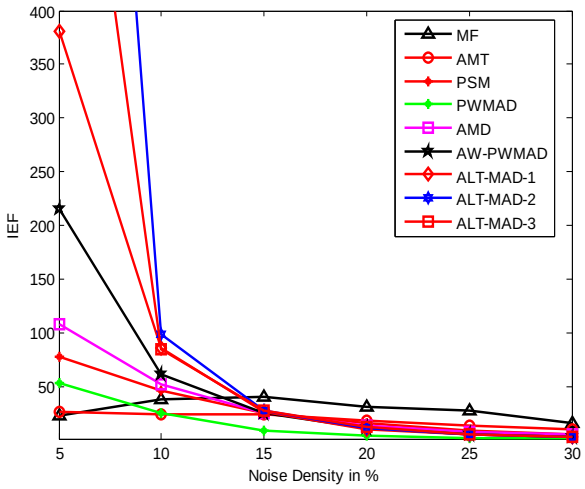
(a) Lena

(b) Pepper

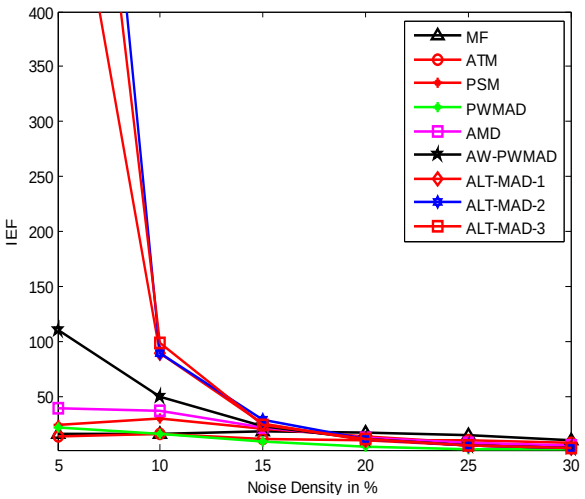
(c) Boat



(a)



(b)



(c)

Fig. 4.6 Performance comparison of various filters in terms of IEF for RVIN at different noise densities on the images:

(a) Lena

(b) Pepper

(c) Boat

Fig. 4.7 Performance of various filters for Lena image with noise density 10% (RVIN)
(a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) TSM (f) PSM (g) PWMAD (h) AMAD (i) AW-PWMAD (j) ALT-MAD-1(k) ALT-MAD-3

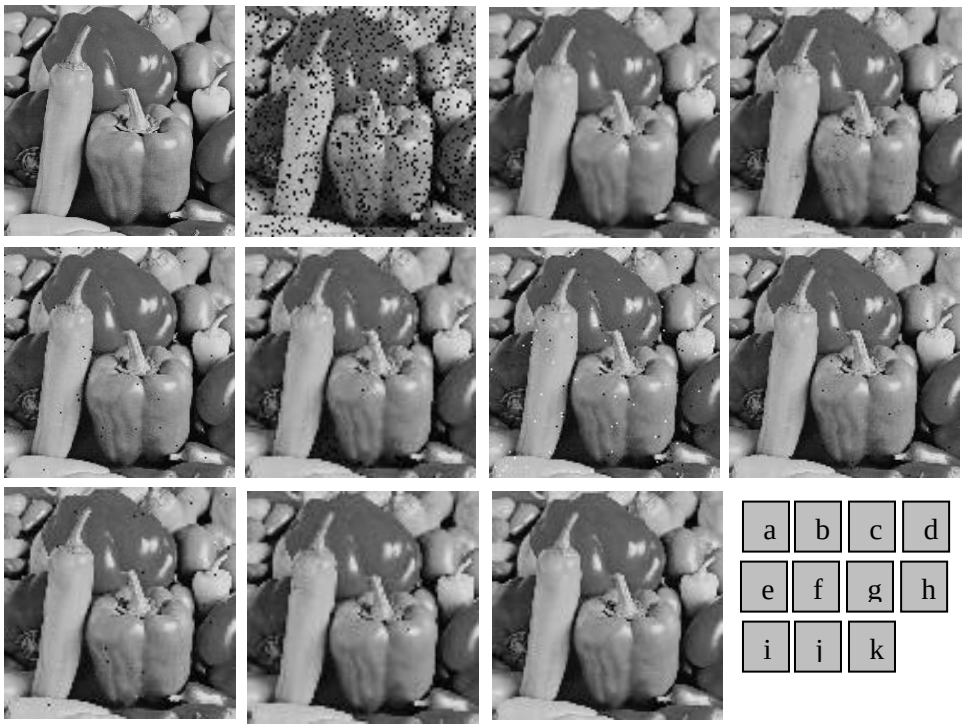


Fig. 4.8 Performance of various filters for Pepper image with noise density 10% (RVIN)
(a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) TSM (f) PSM (g) PWMAD (h) AMAD (i) AW-PWMAD(j) ALT-MAD-1(k) ALT-MAD-3

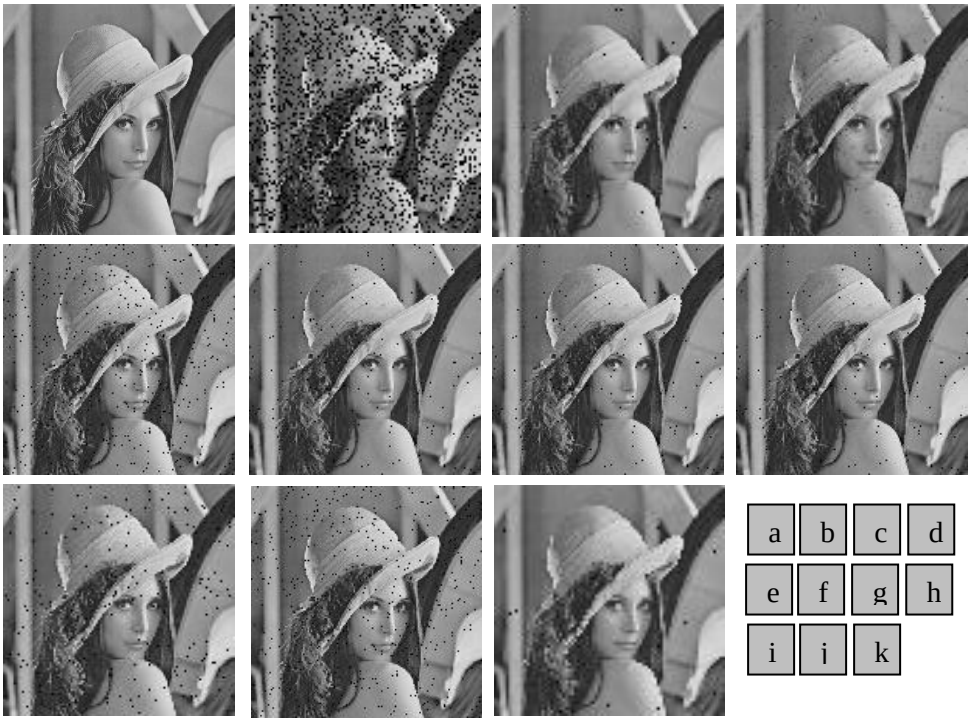


Fig. 4.9 Performance of various filters for Lena image with noise density 20% (RVIN) (a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) TSM (f) PSM (g) PWMAD(h) AMAD (i) AW-PWMAD(j) ALT-MAD-1(k) ALT-MAD-3

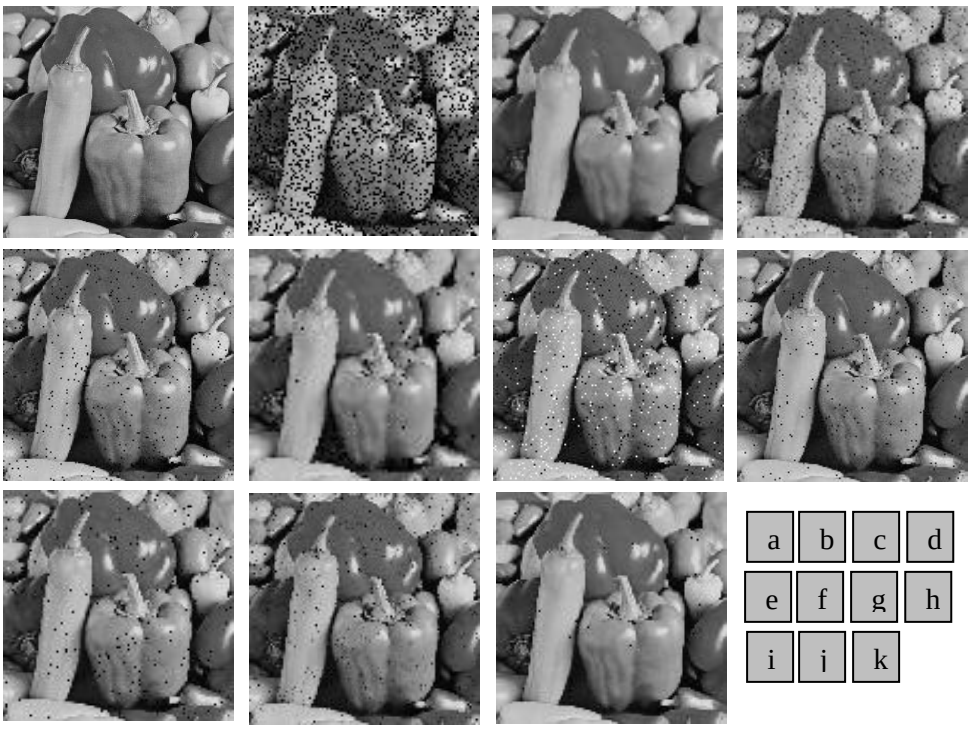


Fig. 4.10 Performance of various filters for Pepper image with noise density 20% (RVIN) (a) Original image (b) Noisy image; Filtered output of: (c) MF (d) ATM (e) TSM (f) PSM (g) PWMAD(h) AMAD (i) AW-PWMAD(j) ALT-MAD-1(k) ALT-MAD-3

4.5 Conclusion

It is observed from the simulation results that the proposed filters are quite effective in suppressing both salt-and-pepper and random-valued impulse noise.

The PSNR tables clearly indicate that the proposed filter ALT-MAD-3 outperform the other existing filters. The filter ALT-MAD-2 shows better performance for low-noise density for pepper image, whereas the filter ALT-MAD-1 is the second best filter. Even the MSE table justifies the same analysis.

From Figs. 4.7 through 4.10 it is observed that the proposed filters are very good in persevering the edges and fine details of an image as compared to other filters. The image quality is evaluated in terms of UQI. From the tables it is observed that the proposed filters show better UQI values than the existing filtering techniques.

Figs. 4.3 through 4.6 show the graphical representation of PSNR, MSE, UQI and IEF. The graphs quickly review the results. They show the performance of filters at various levels of noise densities.

From the results tabulated in Table 4.17, it can be concluded that the proposed filters are very effective in suppression of SPN and RVIN.

The performance of a filter depends on its ability to identify a noisy pixel and replace it with an accurate estimation. It is observed that the proposed filter ALT-MAD-3 shows superior ability to identify a noisy pixel and replace it with quite an accurate estimated value. This is so because this algorithm yields very high PSNR i.e., the error in estimation is very low which indicates high accuracy of the estimation technique.

Chapter 5

Development of Some Color Image Denoising Filters for Suppression of Impulse Noise

5

Preview

Recently, some color image denoising filters are reported in literature [34,132-137].They don't exhibit very high performance in suppressing implosive noise. Hence, there is sufficient scope for developing a good color image denoising filter. Efforts are made, in this research work to develop some high-performance color image filters for filtering SPN and RVIN.

In the proposed methods, the switching median filtering scheme can be extended to denoise corrupted color image using the *scalar median* filtering approach as well as the *vector median* filtering approach. In scalar approach each color component is treated as an independent entity and filtering is applied to each channel in different color spaces (e.g., RGB, YCbCr, etc.). The output signals of independent channels will then be combined to form the recovered color image. H.Zhou, *et al* [34] have shown that the RGB and YCbCr color spaces are found to be quite effective color representation spaces for images (2-D) and video (3-D) denoising applications. Since the performance of denoising filters degrades in other color spaces, efforts are made to develop color image denoising filters in RGB color space only in this research work. Further, RGB filters are simple and hence easy to implement. There would be no need of transformation of an image from RGB to any other color space. This is so because image signals generated from cameras are in RGB color space.

In this chapter three filters are proposed for denoising SPN noise and one filter for RVIN.

The organization of this chapter is given below.

- *Color Image Filters*
- *Multi-Channel Robust Estimator based Impulse-Noise Reduction* (**MC-*REIR*)** *Algorithm*
- *Multi-Channel Impulse-Noise Removal by Impulse Classification* **(MC-*IRIC*)**
- *Multi-Channel Iterative Adaptive Switching Filter* **(MC-*IASF*)**
- *Multi-Channel Adaptive Local Thresholding with MAD* **(MC-*ALT-MAD*)** *Algorithm*
- *Simulation Results*
- *Conclusion*

5.1 Color Image Filters

Though color image filters may be of any color space, only RGB color image filters are discussed here as they exhibit high performance. Fig. 5.1 illustrates the structures of RGB scalar and vector filters. In essence, an RGB scalar filter processes a gray image (single-channel signal) in an iterative paradigm to process a color image. On the other hand, a vector image filter, shown in Fig. 5.1 (b), takes the whole color (all the three channels) information. Though these two types of filters differ from each other by their structure, their resultant operations are identical. Therefore, only RGB scalar filters are developed and depicted in this chapter.

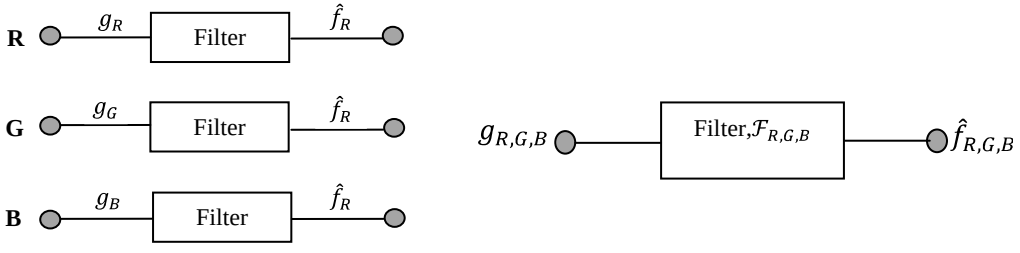


Fig.5.1 Color Image Filters
(a) Scalar filters
(b) Vector Filters

5.2 Multi-Channel Robust Estimator based Impulse-Noise Reduction (MC-REIR) Algorithm

The Robust Estimator based Impulse-Noise Reduction (REIR) Algorithm is a filter that suppresses the SPN from extremely corrupted images. The algorithm is explained in Section 3.2. The filtering performance of this filter is already tested in and it shows better performance. The extension of this filter is presented for suppressing the salt-and-pepper impulse noise in color images. Hence in this chapter algorithm is not explained. The RGB scalar filter structure shown in Fig. 5.1(a) is used for developing of filter. The algorithm is applied separately to R-channel, G- channel and B- channel of noisy image and filtered output of each filter is combined to generate filtered color image.

The performance of this filter is examined by extensive simulation work, and the results are presented in Section-5.6.

5.3 Multi-Channel Impulse-Noise Removal by Impulse Classification (MC-IRIC)

Under high noise density condition, Impulse-Noise Removal by Impulse Classification (IRIC) is a simple and less computational complexity filter that performs very well for suppressing SPN from gray-scale image. The operation of the algorithm is explained in Section 3.4.The performance of the proposed method with gray-scale image is already tested and found to be very promising for all gray-scale test images. Therefore, the application of this method is extended to filter the color images.

The proposed filter is developed by using the RGB scalar filter structure shown in Fig. 5.1(a), where each channel (i.e., R-, G- and B- channel) is filtered separately. The filtered image is restored by combining the filtered output of each channel.

The performance of the filter is tested by extensive simulation and the results are presented in Section-5.6.

5.4 Multi-Channel Iterative Adaptive Switching Filter (MC-IASF)

Iterative Adaptive Switching Filter (IASF) is a high performing filter in suppressing SPN from gray-scale image. The operation of the IASF algorithm is explained in Section 3.6. Due to its iterative structure, the performance of this filter is better than existing order-statistic filters. The performance of this method with gray-scale image

is already tested and the filter is observed to be exhibit very high performance in suppressing SPN under high noise densities. Therefore, the application of this method is extended to filter the color images.

The proposed filter is developed in RGB scalar filters, where each channel is filtered separately. The filtered image is restored by combining the filtered output of each channel.

The performance of filter is tested by extensive simulation and the results are presented in Section 5.6.

5.5 Multi-Channel Adaptive Local Thresholding with MAD (MC-ALT-MAD) Algorithm

The filter, Adaptive Local Thresholding with MAD (ALT-MAD) Algorithm, has better RVIN suppression capability. The operation and performance of the filter is already tested in Section 4.3 of Chapter 4. Hence, this method is extended to filter the random-valued impulse noise in color images. The filter is developed by using RGB scalar filter structure shown in Fig 5.1(a). The algorithm is applied separately to R- channel, G- channel and B- channel and the filtered output of each channel is combined to construct the filtered color image.

The performance of this filter is examined by extensive simulation work. The simulation results are presented in next section.

5.6 Simulation Results

The performance of proposed filters is tested on MATLAB-7.4 platform that sits over a Windows-XP operating system.

The algorithms are tested with different test images. The test images employed here are Lena, Pepper and Tiffany of size 512×512×3, 24-bit color images. These filters are compared with some standard filters. The image metrics: CPSNR, MSE, UQI and IEF are used for performance-evaluation of filters.

The CPSNR values of different filters are presented in tables: Table-5.1throughTable-5.3. MSE values are tabulated in table Table-5.4 through Table-5.6, whereas UQI values are shown in table Table-5.7 through Table-5.9. Tables: Table-5.10 through Table-5.12 present the IEF performance of the filters. The best results are highlighted in bold font for quick analysis in the tables.

The graphical representation of PSNR, MSE, UQI and IEF of the proposed filters and some high performing filters are illustrated in the figures: Fig. 5.2 through Fig. 5.5 for easy analysis. For subjective evaluation, the output images of the proposed and some commonly used filters are shown in the figures: Fig. 5.6 through Fig. 5.9. To show some samples of restored image, for subject evaluation, only Lena and Pepper images corrupted with SPN of noise density 10% and 20% are presented. Conclusions are drawn in next section.

Table-5.1: Performance of various colour image filters in RGB-colour space, in terms of CPSNR (dB) at various noise densities. Test Image: *Lena*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	31.26	26.56	21.20	17.09	13.82	11.13
	MF [5×5]	31.78	31.32	30.14	26.11	21.11	16.05
	MF [7×7]	28.38	28.06	27.60	26.91	25.19	21.64
	ATM	28.23	28.00	27.72	27.28	26.47	24.38
Proposed Filters	MC-REIR	35.03	34.10	32.67	31.11	29.65	27.99
	MC-IRIC	**44.41**	**40.83**	**38.31**	**36.20**	**34.30**	**32.34**
	MC-IASF	34.04	33.05	31.89	30.96	29.78	27.88
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	31.63	26.98	19.53	13.80	10.60	9.07

Table-5.2: Performance of various colour image filters in RGB-colour space, in terms of CPSNR (dB) at various noise densities. Test Image: *Pepper*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	31.28	26.39	21.04	16.87	13.65	11.05
	MF [5×5]	30.90	29.99	28.84	25.54	20.87	16.28
	MF [7×7]	29.76	29.19	28.48	27.50	25.75	21.58
	ATM	29.43	28.99	28.50	27.82	26.85	24.57
Proposed Filters	MC-REIR	34.68	33.45	31.99	30.13	29.01	27.28
	MC-IRIC	**35.13**	**34.17**	**33.64**	**32.17**	**31.60**	**30.28**
	MC-IASF	34.04	33.05	31.89	29.96	28.78	27.88
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	30.91	26.72	19.20	13.45	10.32	8.90

Table-5.3: Performance of various colour image filters in RGB-colour space, in terms of CPSNR (dB) at various noise densities. Test Image: *Tiffany*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	32.94	26.54	20.69	16.47	13.19	10.60
	MF [5×5]	31.78	31.32	30.14	26.46	21.11	16.05
	MF [7×7]	30.41	30.16	29.90	29.42	27.05	22.24
	ATM	30.31	30.18	29.96	29.73	29.07	25.80
Proposed Filters	MC-REIR	37.80	36.60	35.10	33.66	32.19	30.63
	MC-IRIC	**35.46**	**35.20**	**34.73**	**34.11**	**33.34**	**32.44**
	MC-IASF	37.09	35.69	34.25	32.60	30.60	30.45
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	32.49	25.30	15.35	9.18	5.95	4.48

Table-5.4: Performance of various colour image filters in RGB-colour space, in terms of MSE at various noise densities. Test Image: *Lena*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	48.59	143.5	492.9	1267	2696	4771
	MF [5×5]	67.54	78.41	101	188.7	508.6	1468
	MF [7×7]	94.22	101.5	112.8	132.2	183.6	445
	ATM	97.61	102.9	109.7	121.5	146.5	136.8
Proposed Filters	MC-REIR	20.39	25.26	35.16	50.26	70.39	103.2
	MC-IRIC	**2.35**	**5.36**	**9.58**	**15.57**	**24.11**	**37.89**
	MC-IASF	20.89	38.81	45.36	75.05	95.1	118.8
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	44.67	130.1	723.1	2709	5652	8050

Table-5.5: Performance of various colour image filters in RGB-colour space, in terms of MSE at various noise densities. Test Image: *Pepper*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	48.35	150.1	511.4	1335	2799	5103
	MF [5×5]	52.76	65.02	90.80	185.1	531.1	1529
	MF [7×7]	68.67	78.28	92.11	115.6	173.1	451.7
	ATM	73.99	81.94	91.70	107.1	134.2	226.8
Proposed Filters	MC-REIR	22.09	29.32	41.06	62.97	81.63	111.6
	MC-IRIC	**19.93**	**22.68**	**28.08**	**34.79**	**44.90**	**60.94**
	MC-IASF	24.26	33.49	52.91	65.53	85.97	125.9
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	52.63	138.1	780.6	2932	6038	8375

Table-5.6: Performance of various colour image filters in RGB-colour space, in terms of MSE at various noise densities. Test Image: *Tiffany*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	33.03	143.9	554.1	1463	3114	5662
	MF [5×5]	43.13	47.89	62.89	146.7	503.5	1614
	MF [7×7]	59.09	62.61	66.53	74.19	128.1	387.9
	ATM	60.50	62.25	65.56	69.08	80.40	107.9
Proposed Filters	MC-REIR	10.76	14.15	20.07	27.99	39.22	56.22
	MC-IRIC	**18.02**	**19.62**	**21.86**	**25.22**	**30.10**	**37.06**
	MC-IASF	12.00	18.15	34.05	38.92	59.30	77.17
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	36.64	204.0	1895	7837	16549	23144

Table-5.7: Performance of various colour image filters in RGB-colour space, in terms of UQI at various noise densities. Test Image: *Lena*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	**0.986**	0.961	0.880	0.736	0.555	0.387
	MF [5×5]	0.980	0.978	0.972	0.950	0.874	0.702
	MF [7×7]	0.973	0.971	0.969	0.964	0.951	0.889
	ATM	0.972	0.971	0.969	0.966	0.960	0.938
Proposed Filters	MC-REIR	0.647	0.456	0.340	0.254	0.192	0.141
	MC-IRIC	0.653	0.462	0.346	0.265	0.203	0.157
	MC-IASF	0.982	**0.979**	**0.976**	**0.973**	**0.966**	**0.963**
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	0.706	0.692	0.612	0.483	0.4001	0.350

Table-5.8: Performance of various colour image filters in RGB-colour space, in terms of UQI at various noise densities. Test Image: *Pepper*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	0.990	0.971	0.908	0.787	0.626	0.455
	MF [5×5]	0.989	0.987	0.983	0.965	0.903	0.760
	MF [7×7]	0.986	0.985	0.982	0.978	0.968	0.919
	ATM	0.985	0.984	0.982	0.980	0.975	0.958
Proposed Filters	MC-REIR	0.702	0.523	0.399	0.307	0.235	0.174
	MC-IRIC	0.699	0.524	0.403	0.313	0.245	0.189
	MC-IASF	**0.994**	**0.993**	**0.991**	**0.987**	**0.985**	**0.982**
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	0.387	0.380	0.385	0.250	0.184	0.153

Table-5.9: Performance of various colour image filters in RGB-colour space, in terms of UQI at various noise densities. Test Image: *Tiffany*

	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	**0.979**	0.917	0.734	0.517	0.322	0.190
	MF [5×5]	0.972	0.969	**0.960**	0.913	0.755	0.484
	MF [7×7]	0.962	0.960	0.957	0.953	0.922	0.796
	ATM	0.961	0.960	0.958	**0.956**	**0.949**	**0.899**
Proposed Filters	MC-REIR	0.401	0.238	0.159	0.113	0.083	0.059
	MC-IRIC	0.390	0.232	0.161	0.118	0.091	0.073
	MC-IASF	0.938	**0.932**	0.924	0.912	0.892	0.864
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	0.660	0.612	0.459	0.374	0.331	0.306

Table-5.10: Performance of various colour image filters in RGB-colour space, in terms of IEF at various noise densities. Test Image: *Lena*

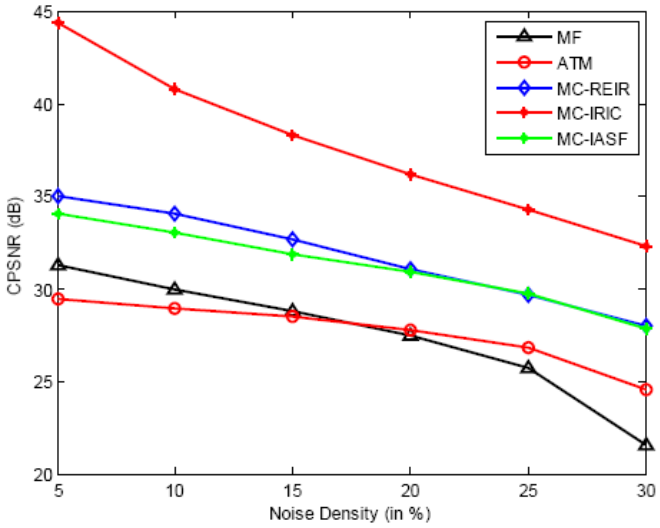
	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	40.69	27.48	12.08	6.24	3.63	2.48
	MF [5×5]	29.37	50.21	58.71	41.92	19.46	8.07
	MF [7×7]	20.98	38.08	52.52	59.98	53.90	26.69
	ATM	20.23	38.51	54.15	65.15	67.66	50.09
Proposed Filters	MC-REIR	96.74	157.8	168.7	157.7	**140.6**	**115.0**
	MC-IRIC	**278.9**	**246.9**	**206.7**	**168.8**	136.7	104.7
	MC-IASF	94.76	144.00	157.07	160.60	127.58	111.03
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	20.96	17.31	5.24	2.01	1.26	1.08

Table-5.11: Performance of various colour image filters in RGB-colour space, in terms of IEF at various noise densities. Test Image: *Pepper*

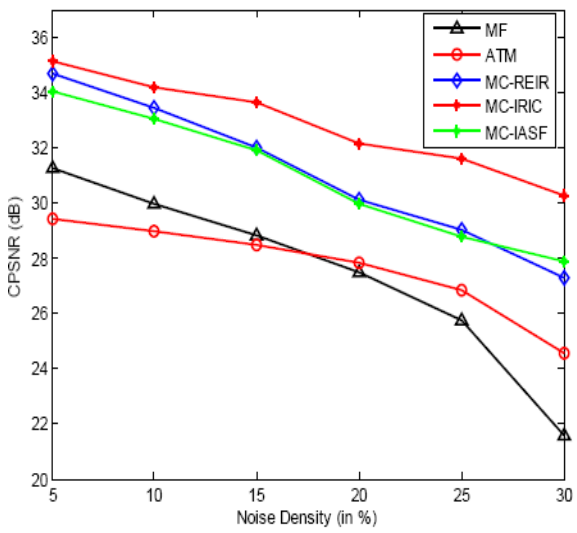
	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	43.00	27.92	12.28	6.26	3.73	2.46
	MF [5×5]	39.65	64.38	69.10	45.37	19.71	8.20
	MF [7×7]	30.61	53.45	67.76	72.12	60.23	27.77
	ATM	28.10	51.27	68.51	77.97	77.70	55.32
Proposed Filters	MC-REIR	**94.10**	**142.9**	**153.1**	**132.9**	**128.1**	103.4
	MC-IRIC	34.89	61.43	74.43	80.36	77.65	68.75
	MC-IASF	64.57	103.2	119.2	127.9	121.7	**118.5**
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	15.85	14.74	4.50	1.75	1.123	0.988

Table-5.12: Performance of various colour image filters in RGB-colour space, in terms of IEF at various noise density conditions. Test Image: *Tiffany*

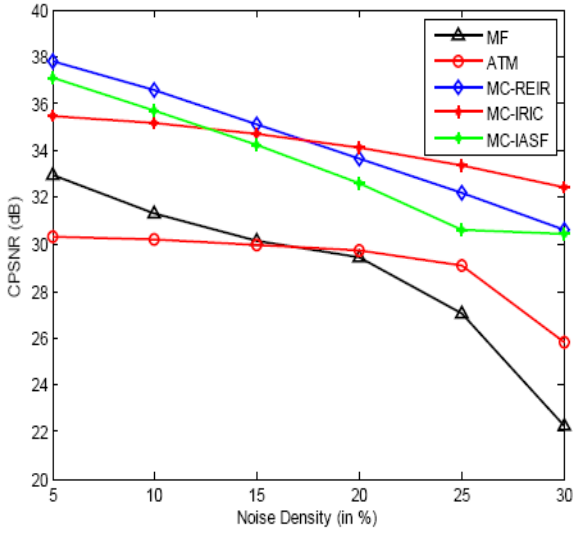
	Filters	% of Noise (Salt-and-pepper Impulse Noise)					
		5	**10**	**15**	**20**	**25**	**30**
Existing Filters	MF [3×3]	70.86	32.68	12.74	6.43	3.77	2.49
	MF [5×5]	54.14	97.87	111.8	64.04	23.27	8.72
	MF [7×7]	39.48	74.86	105.5	126.5	91.71	36.35
	ATM	38.78	75.53	107.3	136.5	145.9	82.40
Proposed Filters	MC-REIR	**96.74**	**157.7**	**168.6**	**157.6**	**140.5**	**115.0**
	MC-IRIC	43.41	80.01	107.0	123.7	130.1	127.1
	MC-IASF	88.40	133.59	145.83	152.62	128.92	105.91
	Random-valued Impulse Noise						
	MC-ALT-MAD-3	57.30	25.98	4.89	1.74	1.08	0.954



(a)



(b)



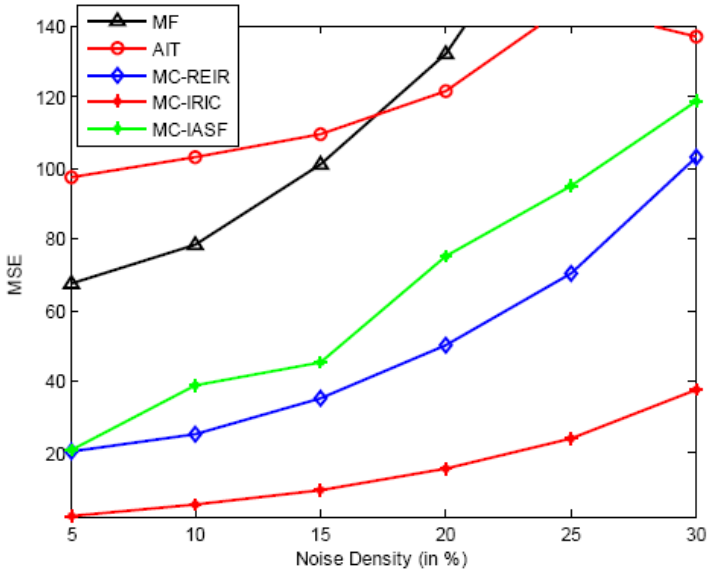
(c)

Fig. 5.2 Performance comparison of various filters in terms of CPSNR (dB) at different noise densities on the images:

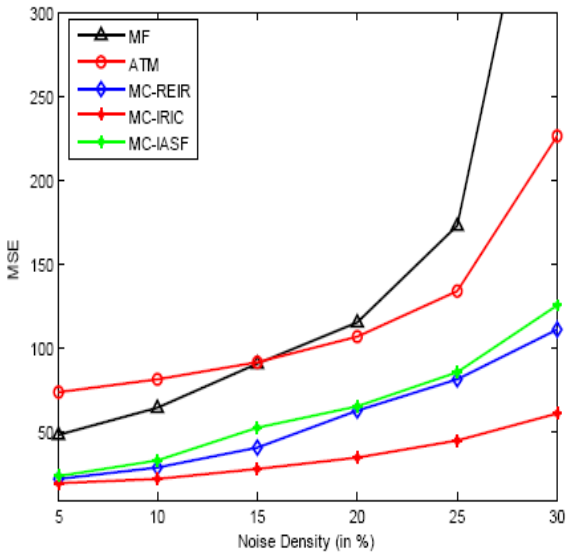
(a) Lena

(b) Pepper

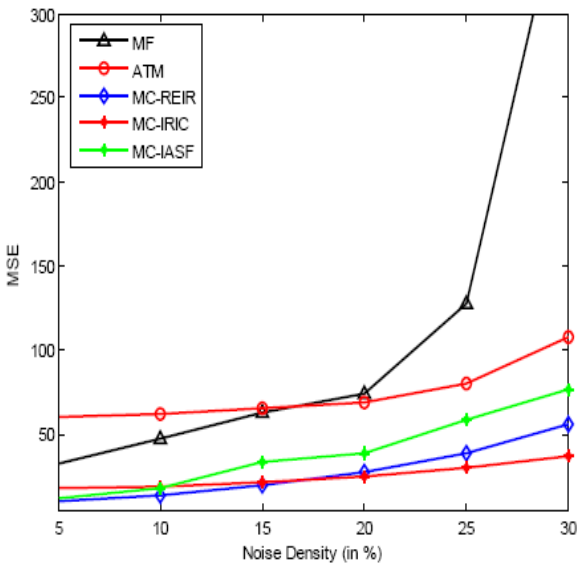
(c) Tiffany



(a)



(b)



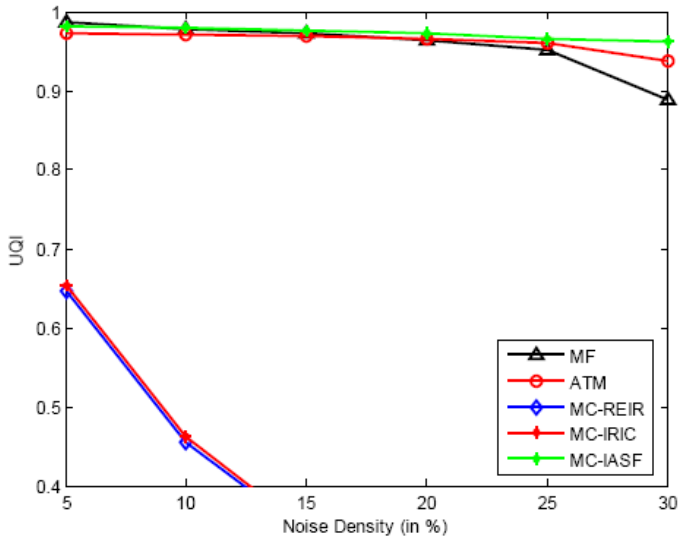
(c)

Fig. 5.3 Performance comparison of various filters in terms of MSE under different noise density on the images:

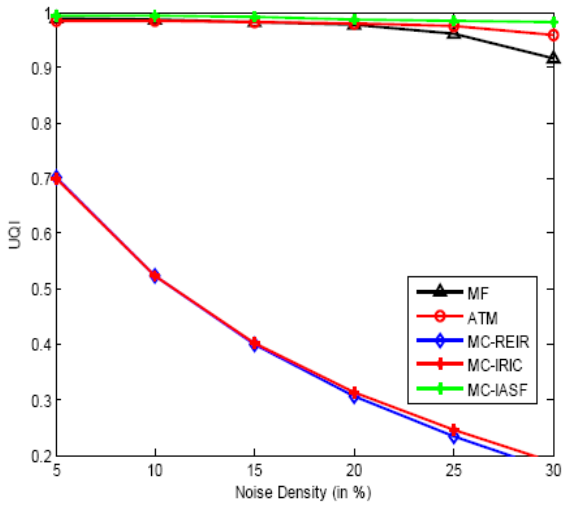
(a) Lena

(b) Pepper

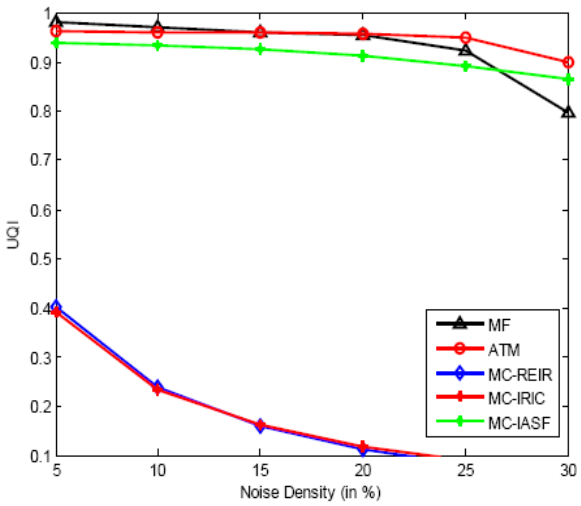
(c) Tiffany



(a)



(b)



(c)

Fig. 5.4 Performance comparison of various filters in terms of UQI under different noise density on the images:

(a) Lena

(b) Pepper

(c) Tiffany

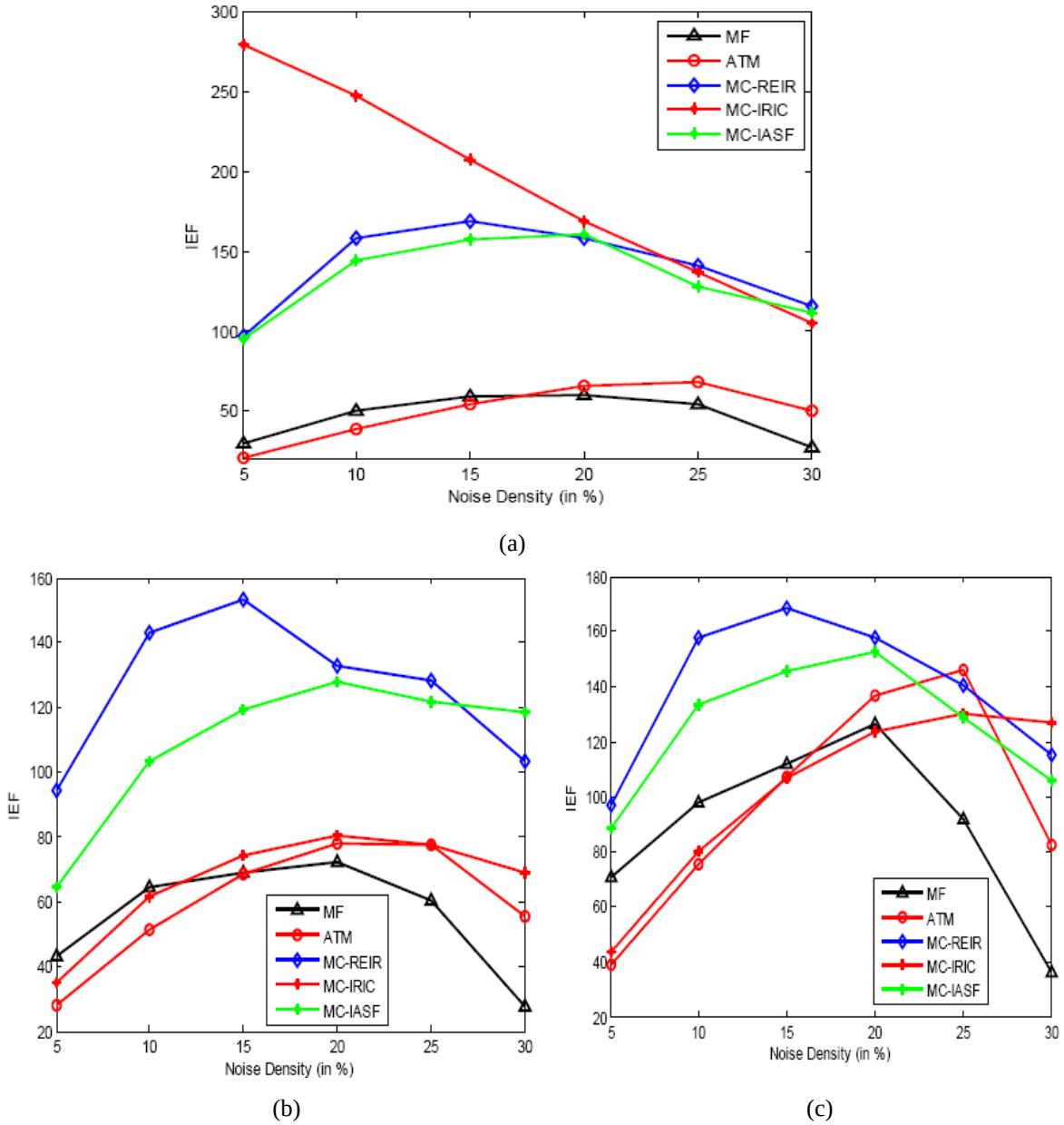


Fig. 5.5 Performance comparison of various filters in terms of IEF under different noise density on the images:

(a) Lena

(b) Pepper

(c) Tiffany

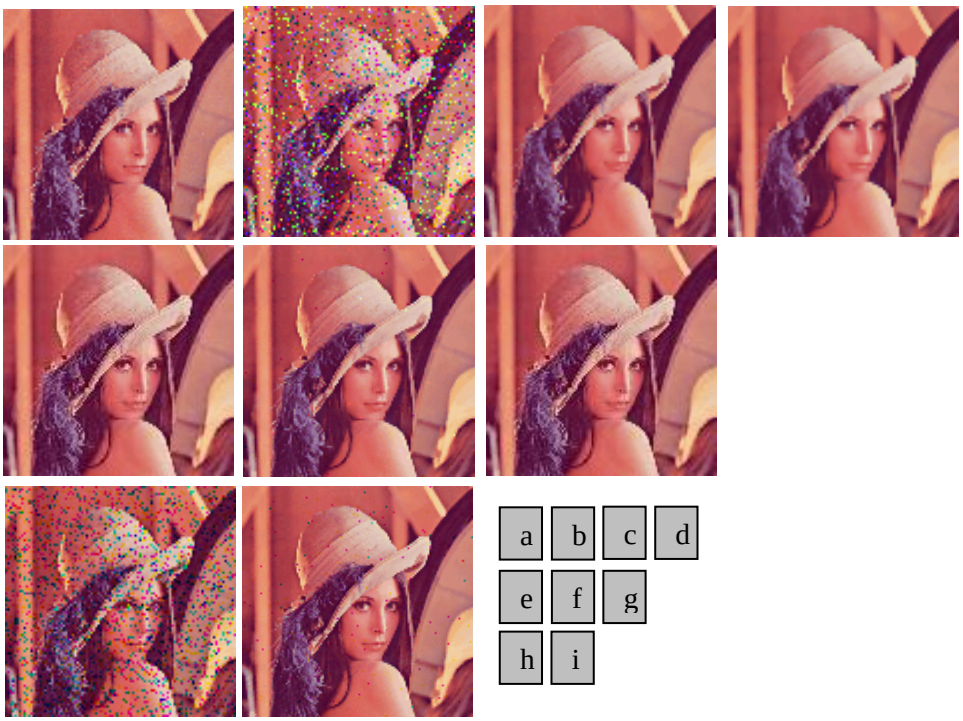


Fig. 5.6 Performance of various filters for Lena image with noise density 10% (a) Original image (b) Noisy image(SPN); Filtered output of: (c) MF (d) ATM (e) MC-REIR (f) MC-IRIC(g) MC-IASF (h) Noisy image(RVIN) ; Filtered output of (i) MC-ALT-MAD-3

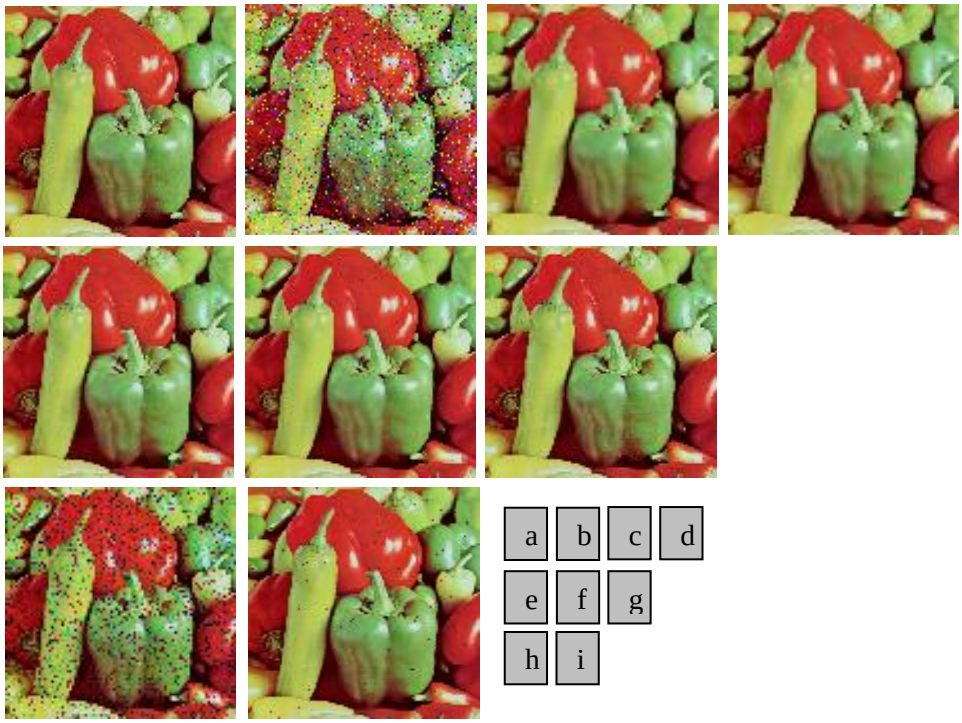


Fig. 5.7 Performance of various filters for Pepper image with noise density 10% (a) Original image (b) Noisy image(SPN); Filtered output of: (c) MF (d) ATM (e) MC-REIR (f) MC-IRIC(g) MC-IASF (h) Noisy image (RVIN) ; Filtered output of (i) MC-ALT-MAD-3

Fig. 5.8 Performance of various filters for Lena image with noise density 20% (a) Original image (b) Noisy image(SPN); Filtered output of: (c) MF (d) ATM (e) MC-REIR (f) MC-IRIC(g) MC-IASF (h) Noisy image(RVIN) ; Filtered output of (i) MC-ALT-MAD-3

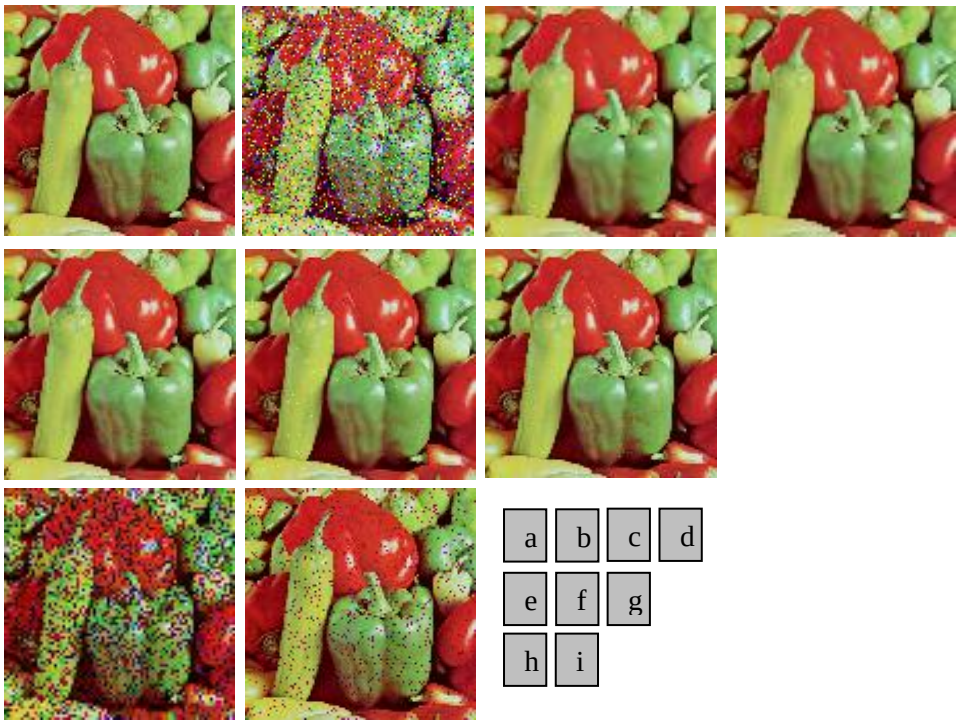


Fig. 5.9 Performance of various filters for Pepper image with noise density 20% (a) Original image (b) Noisy image(SPN); Filtered output of: (c) MF (d) ATM (e) MC-REIR (f) MC-IRIC(g) MC-IASF (h) Noisy image(RVIN) ; Filtered output of (i) MC-ALT-MAD-3

5.7 Conclusion

It is observed from the tables: Table-5.1 through Table-5.3 and tables: Table-5.4 through Table-5.6 that the proposed filter, **MC-IRIC** yields very high CPSNR and very low MSE values under varying noise densities. Therefore, it is really an efficient filter to suppress impulsive noise from color images.

It is observed from the UQI performance tables that for Lena image at low density MF is found to be more effective, whereas for other noise densities the proposed filter, MC-IASF shows better UQI values. For Pepper image, MC-IASF shows better results. Another important observation from UQI table for Tiffany image is that the MF exhibits better results for 5% and 15% of noise and ATM yields better results for higher noise densities.

Further, it is observed from IEF results that, for Lena image, the filter MC-IRIC shows better performance under 5% - 20% noise densities and the filter MC-REIR shows better performance for noise of density more than 25%. For Tiffany image, MC-REIR filter outperforms the other filters; whereas the filter MC-REIR gives the best performance for Pepper image up to 25% of noise densities and MC-IASF shows good results at 30% noise density.

The subjective evaluation should also be taken into consideration to judge the filtering performance. From the figures: Fig. 5.6 through Fig. 5.9 it is observed that for low noise density almost all filters show good image quality. But for 20% noise density MC-IRIC and MC-IASF filters show better quality images with sharp edges. At the same time, MF and ATM filters yield blurring effect which is undesirable.

Thus, it is concluded that:

- MC-MF and MC-ATM are poor performers.
- MC-IASF and MC-REIR exhibit high performance.
- MC-IRIC exhibits the overall best performance.

Chapter 6

Conclusion

6

Preview

The research work in this thesis primarily focuses on impulse noise. Various spatial-domain filters are proposed to suppress the impulse noise, both SPN and RVIN, at very high noise densities. The performances of these filters are compared with the existing filters which are available in literature. Some of them are designed for suppressing SPN while others are meant for RVIN. In addition, some novel filters are developed to suppress impulsive noise from color image. The metrics which are used for performance-analysis are: *peak-signal-to-noise ratio* (PSNR), *universal quality index* (UQI) and *image enhancement factor* (IEF).

The following topics are covered in this chapter.

- Comparative Analysis
- Conclusion
- Scope for Feature Work

6.1 Comparative Analysis

In the first part of this section the performances of filters which are proposed for suppressing impulse noise from gray scale images are analyzed. In the next part, performances of color image filters are analyzed.

6.1.1 Comparative analysis of proposed filters for denoising salt-and-pepper impulse noise in gray-scale images

It is important to test the performance of the filters. The filtered output images are used for analysis purpose. All the filters (existing and proposed) are simulated on MATLAB-7.4 running on a windows-XP operating system.

All the filters are simulated on different test images: *Lena, Pepper* and *Boat*. All test images are of size 512×512 pixels, 8-bit gray-scale images, which are corrupted with salt-and-pepper impulse noise at different noise densities ranging from 10% to 90%.The SPN noise density is categorized as **low** if it is less than 20%, **medium** if noise density is between 20% and 40%, **high** if the noise density is between 40% and 60% and **very high** if noise density more than 60%. Similarly for RVIN noise density is categorized as **low** if the noise density is less than 10%, **medium** if the noise density is between 10% and 20% and **high** if the noise density more than 20%.

The image metrics: PSNR, MSE, UQI, IEF and execution time T_E are used for analysis of filters. To perform the precise comparative study, brief simulation results of all proposed as well as some high-performing existing filters are presented in Table-6.1 for test image Lena. Only PSNR, UQI and IEF are used for quick analysis purpose. The results are presented for 40%, 60% and 80% noise density taking Lena as test image. The best performing filter for each parameter is highlighted in bold font.

It is observed from the table that the proposed filters are performing better in all test parameters (PSNR, UQI and IEF) in all range of noise densities, except the ANDS filter. As it is already analyzed in Chapter-3, ANDS shows good performances only under low noise density conditions.

At noise density of 40%, ASF-I filter shows best result; whereas at a noise density of 60% and above, IASF filter shows its dominating performance.

Table-6.1: Filtering performance of various filters in terms of PSNR (dB), UQI and IEF for SPN Test image: *Lena*

	Noise Densities	40%			60%			80%		
	Filters	PSNR	UQI	IEF	PSNR	UQI	IEF	PSNR	UQI	IEF
Existing filters	MF	27.66	0.975	66.68	25.15	0.955	56.05	20.33	0.869	24.61
	ATM	28.00	0.984	72.34	24.73	0.976	50.76	15.35	0.901	7.81
	CWM	19.21	0.972	9.53	12.48	0.901	3.03	8.19	0.745	1.50
	TSM	18.83	0.835	8.72	12.02	0.466	2.73	7.62	0.132	1.32
	AMF	19.99	0.863	11.34	13.44	0.550	3.74	8.73	0.201	1.69
	PSM	24.83	0.956	35.55	12.26	0.501	2.88	8.11	0.192	1.47
	SMF	24.04	0.944	29.01	17.25	0.771	9.11	9.92	0.298	2.24
	BDND	31.71	0.993	266.4	28.84	0.985	166.1	23.83	0.970	109.8
Proposed filters	ANDS	24.16	0.947	29.60	16.63	0.753	7.89	10.28	0.337	2.42
	REIRA	33.03	0.992	207.5	29.45	0.981	127.1	25.68	0.958	74.79
	IDPSM	30.90	0.988	140.5	27.55	0.974	97.10	24.78	0.952	68.50
	IRIC	32.24	0.991	191.3	28.41	0.979	119.0	23.87	0.941	55.56
	ASF-I	**33.89**	**0.9942**	**279.7**	30.93	0.9885	211.9	27.64	0.9754	132.7
	ASF-II	33.77	0.9940	271.1	29.46	0.9837	151.8	27.24	0.9728	121.0
	IASF	33.78	0.9940	273.5	**31.01**	**0.9887**	**215.7**	**27.75**	**0.9760**	**136.1**

From Table-3.15, it is clear that the proposed filter ASF-I and ASF-II are much more suited for online and real time applications, because of small execution time 1.02 and 0.85 sec. respectively.

6.1.2 Comparative Analysis of Proposed filters for denoising random-valued impulse noise in gray-scale images

In this case also Lena image is used as test image for analysis purpose and PSNR, UQI and IEF are used as image metric. For quick analysis purpose proposed and some existing filters are tested with 10%, 20% and 30% random-valued impulse noise densities. Table-6.2 shows the simulated results. The high performing filters for each parameter is highlighted for quick reference. The overall performances of proposed filters are good in suppressing random-valued impulse noise. From Table-4.17 it is already clear that these filters perform better even in salt-and-pepper impulse noise also.

Table-6.2: Filtering performance of various filters in terms of PSNR (dB), UQI and IEF for RVIN Test image: *Lena*

	Noise Densities	10%			20%			30%		
	Filters	PSNR	UQI	IEF	PSNR	UQI	IEF	PSNR	UQI	IEF
Existing filters	MF[3×3]	29.86	0.985	26.38	20.57	0.890	6.13	14.85	0.679	2.46
	ATM [3×3]	30.02	0.986	24.26	22.22	0.925	8.98	16.15	0.762	3.47
	TSM	30.10	0.988	34.79	21.22	0.921	9.01	15.10	0.725	3.07
	PSM	31.55	0.993	61.03	24.19	0.952	**12.59**	17.36	0.810	4.08
	PWMAD	28.28	0.984	24.35	19.01	0.869	5.11	15.84	0.635	2.09
	AMAD	31.94	0.994	48.48	24.22	0.953	10.86	17.56	0.829	5.96
Proposed filters	AW-PWMAD	33.54	0.993	62.10	23.37	0.938	11.77	17.02	0.776	4.09
	ALT-MAD-1	36.88	0.9970	87.73	25.01	0.9552	11.21	18.54	0.8278	3.87
	ALT-MAD-2	36.21	0.9966	77.46	24.71	0.9545	11.15	18.13	0.8173	3.64
	ALT-MAD-3	**37.02**	**0.9970**	**87.83**	**25.22**	**0.9578**	11.43	**18.98**	**0.8344**	**4.10**

6.1.3 Comparative Analysis of Proposed filters for denoising salt-and-pepper impulse noise in color images

The simulation of color filter is also performed on the similar condition as stated in Section-6.1.1. The test images: *Lena*, *Pepper* and *Tiffany* of size (512×512×3) are used. They are simulated in RGB color space. The performances of these filters are compared with some commonly used filters. Table-6.3 shows the results for analysis purpose. Here color Lena image corrupted with 10%, 20% and 30% SPN is used to test the filtering performance of filters. The best performing filter with parameters are highlighted for reference. It is clear from Table-6.3 that the proposed filters perform better. Out of these, MC-IRIC filter shows its better performance in terms of CPSNR and IEF and at the same time the filter: MC-IASF shows its better performance in terms of UQI only. This indicates that the proposed scheme: MC-IASF yields very low distortion. So, it is clear from the results that the proposed filters suppress noise quite well in color images.

Table-6.3: Filtering performance of various filters in terms of CPSNR (dB), UQI and IEF for SPN. Test image: *Lena* (color)

	Noise Densities	10%			20%			30%		
	Filters	CPSNR	UQI	IEF	CPSNR	UQI	IEF	CPSNR	UQI	IEF
Existing Filters	MF	31.32	0.978	50.21	26.91	0.964	59.98	21.64	0.889	26.69
	ATM	28.00	0.971	38.51	27.28	0.966	65.15	24.38	0.938	50.09
Proposed Filters	MC-REIR	34.10	0.456	157.8	31.11	0.254	157.7	27.99	0.141	**115.0**
	MC-IRIC	**40.83**	0.462	**246.9**	**36.20**	0.265	**168.8**	**32.34**	0.157	104.7
	MC-IASF	33.05	**0.979**	144.0	30.96	**0.973**	160.6	27.88	**0.963**	111

6.2 Conclusion

The simulated results show that the proposed filters suppress the impulse noise better than the existing filters, both in gray-scale as well as in color images.

It is observed from Table-6.1 that the proposed filters perform better than the existing methods for suppression of SPN. The proposed methods: **ASF-1** and **IASF** exhibit quite superior performance compared to other methods as they yield high PSNR, UQI and IEF values. The performance of a filter depends on its ability to identify a noisy pixel and replace it with an efficient estimation. The IASF algorithm is iterative in nature which makes it more efficient in proper noise detection. Further, in both the algorithms, adaptive filtering window helps to retain the edges and fine details of an image. Hence, these two filters show better noise suppressing capability without yielding any appreciable distortion and blur.

For online and real-time applications the system must have small execute time with less complexity. Table-3.21 shows the filter **ASF-I** and **ASF-II** shows its best performance along with good filtering operation.

Similarly, from Table-6.2, it is observed that the proposed methods are efficient in suppressing RVIN. Out of all, **ALT-MAD-3** exhibits much better performance in terms of PSNR, UQI and IEF compared to other proposed and existing methods.

In color filtering, the proposed filter **MC-IRIC** shows its best performance.

Finally, it may be concluded that ASF-I and IASF excel in suppressing high density SPN whereas ALT-MAD-3 yields excellent performance in filtering RVIN of low and medium densities. The proposed filter MC-IRIC is the best candidate for color image denoising.

6.3 Scope for Future Work

There are some new directions of research in the field of image denoising which are yet to explore completely. So, there is sufficient scope to develop very efficient filters in the direction mentioned below.

i) Fuzzy logic and neural network may be used for perfect classification of noisy pixels which can improve the filtering performance.

ii) The window size and shape of filter can be made adaptive for different or multiple parameters.

REFERENCES

[1] R.C. Gonzalez and R.E. Woods, Digital Image Processing. 2nd ed. Englewood Cliffs, NJ: Prentice-Hall; 2002.

[2] A.K. Jain, Fundamentals of Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall; 1989.

[3] Scott E Umbough, DigitalnImage Processing and Analysis – Human and Computer Vision Applications with CVIP tools, Second Edition, CRC press, Taylor & Francis group, 2011.

[4] J. Astola and P. Kuosmanen, Fundamentals of Nonlinear Digital Filtering. Boca Raton, FL: CRC Press, 1997.

[5] I. Pitas and A.N. Venetsanopoulos, Nonlinear Digital Filters: Principles and Applications. Norwell, MA: Kluwer, 1990.

[6] W.K. Pratt, Digital Image Processing, NY: John Wiley & Sons, 2001.

[7] R. Lukac and K.N. Plataniotis, Color Image Processing: Methods and Applications, Boca Raton, FL: CRC Press, 2007.

[8] J.S. Lim, Two-Dimensional Signal and Image Processing. Englewood Cliffs, NJ: Prentice-Hall; 1990.

[9] S. Mallat, A Wavelet Tour of Signal Processing, 2nd ed. Academic Press, Inc., 1999.

[10] G. Strang and T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, 1996.

[11] J.C. Goswami and A.K. Chan, Fundamentals of Wavelets: Theory, Algorithms and Applications, Wiley, 1999.

[12] Y. Meyer, R. D. Ryan, Wavelets: Algorithms & Applications, Society for Industrial and Applied Mathematics, Philadelphia, 1993.

[13] I. Daubechies, Ten Lectures on Wavelets, Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[14] K.R. Rao and P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications, Academic Press, Boston, 1990.

[15] Norbert Wiener, Extrapolation, Interpolation and Smoothing of Stationary Time Series, New York: Wiely, 1949.

[16] N. Ahmed, T. Natarajan and K.R. Rao, "Discrete Cosine Transform," IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90-93, 1974.

[17] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no.7, pp. 674-692, 1989.

[18] M. Vetterli, C. Herley, "Wavelets and Filter Banks: Theory and Design," IEEE Transactions on Signal Processing, vol. 40, no. 9, pp. 2207-2232, 1992.

[19] D. Stanhill, Y.Y. Zeevi, "Two-Dimensional Orthogonal and Symmetrical Wavelets and Filter-Banks," IEEE Acoustics, Speech and Signal Processing, vol. 3, pp. 1499-1502, 1996.

[20] M.Mastriani, A.E. Giraldiz, "Smoothing of Coefficients in Wavelet domain for Speckle reduction in Aperature Radar Images," GVIP Special issue, pp 1-7, Dec-2007.

[21] R.N. Bracewell, "Discrete Hartley transform, " Journal of Opt. Soc. America, vol. 73, no. 12, pp. 1832-1835, 1983.

[22] T. Cooklev, A. Nishihara, "Biorthogonal coiflets," IEEE Transactions on Signal Processing, vol. 47, no. 9, pp. 2582-2588, 1999.

[23] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, 1986.

[24] L S. Davis, "A Survey on Edge Detection Techniques," Computer Graphics and Image Processing, vol. 4, pp. 248-270, 1975.

[25] F. Russo, "Color Edge Detection in Presence Of Gaussian Noise Using Nonlinear Pre-Filtering," IEEE Transactions on Instrumentation and Measurement, vol. 54, no. 1, pp. 352-358, 2005.

[26] P. Bao, L. Zhang, X. Wu, "Canny Edge Detection Enhancement by Scale Multiplication," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 9, pp. 1485-1490, 2005.

[27] P. Hsiao, C. Chen, H. Wen, S. Chen, "Real-Time Realization of Noise-Immune Gradient-Based Edge Detector," Proc. IEE Computers and Digital Techniques, vol. 153, no. 4, pp. 261-269, 2006.

[28] T. Chaira, A.K. Ray, "A New Measure Using Intuitionist Fuzzy Set Theory And Its Application To Edge Detection," Applied Soft Computing, vol. 8, no. 2, pp. 919-927, 2008.

[29] P. S. Windyga, "Fast Impulsive Noise Removal", IEEE Trans. on image Processing, vol.10, no. 1, pp.173-179, January, 2001.

[30] F. Russo, "Evolutionary Neural Fuzzy System For Noise Cancellation In Image Processing", IEEE Trans. Inst & Meas., vol. 48, no. 5, pp. 915-920, Oct. 1999.

[31] F. Farbiz, M. B. Menhaj, S.A. Motamedi and M.T. Hagan," A New Fuzzy Logic Filter For Image Enhancement", IEEE Trans. on Syst. Meas. Cybernetics – Part B: Cybernetro, vol.30, no.1, pp. 110-119, Feb 2000.

[32] K.N. Plataniotis, S. Vinayagamoorthy, D. Androutsos, A.N. Venetsanopoulos, "An Adaptive Nearest Neighbor Multichannel Filter", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 6, pp. 699-703, 1996.

[33] S. Schulte, S. Morillas, V. Gregori, E. Kerre, "A New Fuzzy Color Correlated Impulse Noise Reduction Method," IEEE Transactions on Image Processing, vol. 16, no. 10, pp. 2565-2575, 2007.

[34] H. Zhou, K.Z. Mao, "An Impulsive Noise Color Image Filter Using Learning-Based Color Morphological Operations," Digital Signal Processing, vol. 18, pp. 406-421, 2008.

[35] Y. Li, G.R. Arce, J. Bacca, "Weighted Median Filters For Multichannel Signals," IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4271-4281, 2006.

[36] S. Meher, G. Panda, B. Majhi, M.R. Meher, "Fuzzy Weighted Rank-Ordered Mean (FWROM) Filters For Efficient Noise Suppression", Proc. of National Conference on Communication, NCC-2005, Indian Institute of Technology, Kharagpur, 2005.

[37] S. Meher, "Development of Some Novel Nonlinear and Adaptive Digital Image Filters for Efficient Noise Suppression", Doctoral Dissertation, National Institute of Technology, Rourkela, India.

[38] R. Garnett, T. Huegerich, C. Chui, W. He, "A Universal Noise Removal Algorithm With An Impulse Detector," IEEE Transactions on Image Processing, vol.14, no. 11, pp. 1747-1754, 2005.

[39] R. Li, Y.J. Zhang, "A Hybrid Filter For The Cancellation Of Mixed Gaussian Noise And Impulse Noise," Proceedings of the Joint Conference of the Fourth International Conference on Information, Communications and Signal Processing and Fourth Pacific Rim Conference on Multimedia, ICICS-PCM, 2003.

[40] I. H. Jang and N. C. Kim, "Locally Adaptive Wiener Filtering In Wavelet Domain For Image Restoration," Proc. IEEE R-10 Conference on Speech and Image Technologies for Computing and Telecommunications, TENCON-97, vol. 1, pp. 25-28, 1997.

[41] G. Angelopoulos and I. Pitas, "Multichannel Wiener Filters In Color Image Restoration," IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 1, pp.87-102, 1994.

[42] L. Shao, R. M. Lewitt, J.S. Karp and G. Muehllehner, "Combination Of Wiener Filtering and Singular Value Decomposition Filtering for Volume Imaging PET," IEEE Transactions on Nuclear Science, vol. 42, no. 4, pp. 1228-1234, 1995.

[43] J.S. Lee, "Digital Image Enhancement And Noise Filtering by Use Of Local Statistics," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 3, pp. 165-168, 1980.

[44] P. Perona and J. Malik, "Scale Space And Edge Detection Using Anisotropic Diffusion," in Proc. IEEE Comput. Soc. Workshop Comput. Vision, Miami, FL, pp. 16-22, 1987.

[45] P. Perona and J. Malik, "Scale Space And Edge Detection Using Anisotropic Diffusion," IEEE Trans. Pattern Anal. Machine Intell., vol. 12, no. 7, pp. 629-639, 1990.

[46] L. I. Rudin, S. Osher and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," Physica D, vol. 60, pp. 259-268, 1992.

[47] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images," Proceedings of IEEE international conference on computer vision, pp. 839-846, 1998.

[48] N.Bhoi, S. Meher, "Circular Spatial Filtering Under High Noise Variance Conditions," Computers & Graphics, Elsevier, vol. 32, no. 5, pp. 568-580, 2008.

[49] F. Durand, J. Dorsey, "Fast Bilateral Filtering For The Display Of High-Dynamic-Range Images," ACM Transactions on Graphics, vol. 21, no. 3, pp. 257-266, 2002.

[50] A. Buades, B. Coll, and J. Morel, "A Non-Local Algorithm For Image Denoising," Proc. IEEE international conference on computer vision and pattern recognition, pp. 60-65, 2005.

[51] T. Rabie, "Robust Estimation Approach for Blind Denoising", IEEE Transactions on Image Processing, vol. 14, no. 11, pp. 1755-1765, 2005.

[52] Z. Hou, T.S. Koh, "Image Denoising Using Robust Regression," IEEE Signal Processing Letters, vol. 11, no. 2, pp. 243-246, 2004.

[53] H. Takeda, S. Farsiu, P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," IEEE Transactions on Image Processing, vol. 16, no. 2, pp.349-365, 2007.

[54] D. V. Ville, M. Nachtegael, D.V. Weken, E.E. Kerre, W. Philips, I. Lemahieu, "Noise Reduction by Fuzzy Image Filtering," IEEE Transactions on Fuzzy Systems, vol.11, no. 4, pp. 429-436, 2003.

[55] E. Bala, A. Ertüzün, "A Multivariate Thresholding Techniques for Image Denoisng Using Multi-Wavelets," EURASIP Journal an Applied Signal Processsing, vol. 8, pp. 1205-1211, 2005.

[56] R. Eslami, H. Radha, "Translation-Invariant Contourlet Transform and Its Applications to Image Denoising," IEEE Transactions on Image Processing, vol. 15, no. 11, pp. 3362-3374, 2006.

[57] P. Scheunders, S. Backer, "Wavelet Denoising of Multi-component Images Using Gaussian Scale Mixtures Models and A Noise-Free Images As Priors," IEEE Transactions on Image Processing, vol. 16, no. 7, pp. 1865-1872, 2007.

[58] D. R. K. Brownrigg, "The Weighted Median Filter," Comm. ACM, vol. 27, pp. 807-818, August 1984.

[59] Marco Fischer, Jose L.Paredes and Gonzalo.R. Arce, "Weighted Median Image Sharpeners for the Word Wise Web." IEEE Transactions on Image Processing, vol. 11, no. 7, pp. 717-727,July 2002

[60] B. I. Justusson, "Median Filtering: Statistical Properties," Two-Dimensional Digital Signal Processing II, T. S. Huang Ed., New York; Springer Verlag, 1981.

[61] T. Loupos, W. N. McDicken and P. L. Allan, "An Adaptive Weighted Median Filter for Speckle Suppression In Medical Ultrasonic Images," IEEE Trans. Circuits Syst., vol. 36, no.1, pp. -129-135, Jan. 1989.

[62] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted Median Filters: A Tutorial," IEEE Trans. Circuits and Syst. II : Analog and Digital Signal Processing, vol. 43, pp. 157-192, Mar. 1996.

[63] T.C. Ayasal and K.E. Barner, "Quadratic Weighted Median Filters for Edge Enhancement of Noisy Image," IEEE Transactions on Image Processing, vol. 15, no. 11, pp. 3294-3301, Nov - 2006.

[64] S-J. Ko and Y. H. Lee, "Center-Weighted Median Filters and Their Applications to Image Enhancement," IEEE Trans. Circuits and Syst., vol. 38, no. 09, pp. 984-993, Sept. 1991.

[65] Arce.G. and Paredes.J, "Recursive Weighted Median Filters Admitting Negative Weights and Their Optimization," IEEE Trans. on Signal processing, Vol. 48, no. 3, pp. 768 – 779, 2000.

[66] T. Chen and H.R. Wu, "Adaptive Impulse Detection Using Center-Weighted Median Filters." IEEE Signal Processing Letters, vol. 8, no. 1, pp.1-3, Jan 2001.

[67] J. B. Bednar and T. L.Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 1, pp. 145–153, Feb. 1984.

[68] A. C. Bovik, T. Huang and D. C. Munson, "A Generalization Of Median Filtering Using Linear Combinations of Order Statistics," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-31, pp. 1342-1350, Jun. 1983

[69] D. A. Florencio and R. W. Schafer, "Decision-Based Median Filter Using Local Signal Statistics," in Proc. SPIE Vis. Commun. Image Process., vol. 23, no.8, pp. 268–275, Sep. 1994.

[70] P. S. Windyga, "Fast Impulse Noise Removal," IEEE Trans. *Image Process.*, vol. 10, no. 1, pp. 173–179, Jan. 2001.

[71] N. Alajlan, M. Kamel, and E. Jernigan, "Detail Preserving Impulsive Noise Removal," Signal Process. Image Commun., vol. 19, pp. 993–1003, 2004.

[72] Z. Wang and D. Zhang, "Progressive Switching Median Filter For The Removal of Impulse Noise From Highly Corrupted Images," IEEE Trans. Circuits Syst. II, vol. 46, no. 1, pp. 78–80, Jan. 1999.

[73] S. Zhang and M. A. Karim, "A New Impulse Detector For Switching Median Filters," IEEE Signal Process. Lett., vol. 9, no. 4, pp. 360–363, Nov. 2002.

[74] H.L. Eng and K.K. Ma, "Noise Adaptive Soft-Switching Median Filter," IEEE Trans. Image Process., vol. 10, no. 2, pp. 242–251, Feb. 2001.

[75] H. Hwang and R. A. Haddad, "Adaptive Median Filters: New Algorithms And Results," IEEE Trans. Image Process., vol. 4, no. 4, pp. 499–502, April 1995.

[76] G. Pok, J.C. Liu, and A. S. Nair, "Selective Removal Of Impulse Noise Based On Homogeneity Level Information," IEEE Trans. Image Processing, vol. 12, no. 1, pp. 85–92, Jan. 2003.

[77] A. Hamza and H. Krim, "Image Denoising A Nonlinear Robust Statistical Approach." IEEE. Trans. Signal Processing, vol. 49, no. 12, pp. 3045-3054, 2001.

[78] M. Black and A. Rangarajan, "On the Unification of Line Processes, Outlier Rejection, and Robust Statistics with Applications To Early Vision,"International Journal of Computer Vision, vol.19, pp. 57-91, 1996.

[79] H-L Eng and K-K Ma, " Noise Adaptive Soft Switching Median Filter", IEEE Trans. on Image Processing, vol.10, no.2, pp. 242-251, Feb 2001.

[80] W.-Y. Han and J.-C. Lin, "Minimum–Maximum Exclusive Mean (MMEM) Filter To Remove Impulse Noise From Highly Corrupted Images," Electron. Lett., vol. 33, no. 2, pp. 124–125, 1997.

[81] J.-H. Wang, "Pre-scanned Min-Max Center-Weighted Filters For Image Restoration," Proc. Inst. Elect. Eng., vol. 146, no. 2, pp. 101–107, 1999.

[82] C.-T. Chen and L.-G. Chen, "A Self-Adjusting Weighted Median Filter For Removing Impulse Noise In Image," in Proc. IEEE Int. Conf. on Image Processing, pp. 419–422, 1998.

[83] T. Kasparis, N. S. Tzannes, and Q. Chen, "Detail-Preserving Adaptive Conditional Median Filters," J. Electron. Imag., vol. 1, no. 14, pp. 358–364, 1992.

[84] A. Sawant, H. Zeman, D. Muratore, S. Samant, and F. DiBianka, "An Adaptive Median Filter Algorithm To Remove Impulse Noise In X-Ray And CT Images And Speckle In Ultrasound Images," *Proc. SPIE*, vol. 3661,pp. 1263–1274, Feb. 1999.

[85] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A Signal-Dependent Rank Ordered Mean Filter-A New Efficient Approach For The Removal Of Impulse Noise From Highly Corrupted Images," IEEE Trans. ImageProcess., vol. 5, no. 6, pp. 1012–1025, Jun.1996.

[86] T. Chen, K. K. Ma and L. H. Chen, "Tri-State Median Filter For Image Denoising", IEEE Trans. on Image Processing, vol. 8,no.12, pp.1834-1838, December, 1999.

[87] X. Xu, E. L. Miller, D. Chen, and M. Sarhadi, "Adaptive Two-Pass Rank Order Filter To Remove Impulse Noise In Highly Corrupted Images," IEEE Trans. Image Process., vol. 13, no. 2, pp. 238–247, Feb. 2004.

[88] M. Nikolova, "A Variational Approach To Remove Outliers And Impulse Noise," Journal of Math. Imag. Vis., vol. 20, pp. 99–120, 2004.

[89] R. H. Chan, C. Hu, and M. Nikolova, "An Iterative Procedure For Removing Random-Valued Impulse Noise," IEEE Signal Process. Lett., vol. 11, no. 12, pp. 921–924, Dec. 2004.

[90] I. Aizenberg, C. Butakoff, and D. Paliy, "Impulsive Noise Removal Using Threshold Boolean Filtering Based On The Impulse Detecting Functions," IEEE Signal Process. Lett., vol. 12, no. 1, pp. 63–66, Jan. 2005.

[91] J. H. Wang and L. D. Lin, "Image Restoration Using Parametric Adaptive Fuzzy Filter," in Proc. NAFIPS, vol. 1, pp. 198–202, 1998.

[92] J. H.Wang and H. C. Chiu, "HAF: An Adaptive Fuzzy Filter For Restoring Highly Corrupted Images By Histogram Estimation," Proc. Nat. Sci. Council, vol. 23, no. 5, pp. 630–643, 1999.

[93] F. Russo and G. Ramponi, "A Fuzzy Filter For Images Corrupted By Impulse Noise," IEEE Signal Process. Lett., vol. 3, no. 6, pp. 168–170, Jun. 1996.

[94] D. Van De Ville, M. Nachtegael, D. Van der Weken, E. E. Kerre, W. Philips, and I. Lemahieu, "Noise Reduction by Fuzzy Image Filtering," IEEE Trans. Fuzzy Syst., vol. 11, no. 4, pp. 429–436, Aug. 2003.

[95] H. Xu, G. Zhu, H. Peng, and D. Wang, "Adaptive Fuzzy Switching Filter For Images Corrupted By Impulse Noise," Pattern Recognit. Lett., vol. 25, pp. 1657–1663, 2004.

[96] Y. S. Choi and R. Krishnapuram, "A Robust Approach To Image Enhancement Based On Fuzzy Logic," IEEE Trans. Image Process., vol. 6, no. 6, pp. 808–825, Jun. 1997.

[97] M. E. Yüksel and A. Bas¸türk, "Efficient Removal Of Impulse Noise From Highly Corrupted Digital Images by A Simple Neuro-Fuzzy Operator," Int. Journal Electronics Communication, vol. 57, no. 3, pp. 214–219, 2003.

[98] D. Zhang and Z. Wang, "Impulse Noise Detection And Removal Using Fuzzy Techniques," Electron. Lett., vol. 33, no. 5, pp. 378–379, 1997.

[99] K. Arakawa, "Median Filters Based On Fuzzy Rules And Its Application To Image Restoration," Fuzzy Sets Syst., vol. 77, pp. 3–13, 1996.

[100] GwanggilJe on, Yong Fang,Sang-JunPark Rokkyu lLee, and Jechang Jeong, "Cascade Fuzzy Filter For Impulse And Random Noise Concellation,"3ed IEEE International Symposium on Consumer Electronic – 2009, pp. 498-502, 2009.

[101] Yuksel,M.E. ;Besdok.E, "A Simple Neuro-Fuzzy Impulse Detection For Efficient Blur Reduction of Impulse Noise Removal Operations For Digital Images", IEEE Trans. on Fuzzy system, vol. 12, no. 6, pp. 854-865, Dec-2004.

[102] Hao Qin, Simon.X.Yang, "Adaptive Nero-Fuzzy Inference System Based Approach to Non Linear Noise Cancellation for Images", Fuzzy sets and systems, vlo. 158, no. 16, pp. 1036-1063, May-2007.

[103] Yuksel,M.E, "A Hybrid Nero-Fuzzy Filter For Edge Preserving Restoration of Images Corrupted By Impulse Noise", IEEE Signal Processing, vol. 15, no. 4, pp. 928-936,Apel-2006.

[104] Chang-Shing Lee, Shu-Mei Guo, Chin-Yuan Hus, "Genetic-Based Fuzzy Image Filter and Its Application to Image Processing", IEEE Trans. On systems, Man, and Cybernetics, vol. 35, no. 4, pp. 1083-4419,Aug-2005.

[105] Yuksel,M.E. ;Besturk.A, " Efficient Removal of Impulse Noise form Highly Corrupted Digital Image by Simple neuro-fuzzy Operator', Int. Journal of Electronics and Commun., vol. 57, no. 3, pp. 214-219, 2003.

[106] Pei-Eng Ng and Kai-Kuang Ma, "A Switching Median Filter With Boundary Discriminative Noise Detection For Extremely Corrupted Images"IEEE Trans. ImageProcessing, vol. 15, no. 6, pp. 1506–1516, June 2006.

[107] A. C. Bovik, T. S. Huang, and D. C. Munson, "A Generalization Of Median Filtering Using Linear Combinations of Order Statistics," IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-31, no. 6, pp. 1342–1350, Dec. 1983.

[108] W. Luo, "An Efficient Detail-Preserving Approach For Removing Impulse Noise In Image," IEEE Trans. Signal Process., vol. -13, no. 7, pp. 413–416, July 2006.

[109] Y. Wang, Linda S. DeBrunner, J. P. Havlicek, and D. Zhou, "Signal exclusive adaptive average filter for impulse noise suppression", 2006 IEEE Southwest Symposium on Image Analysis and Interpretation, *pp. 51–55, 2006.*

[110] S.Saudia, S.Annam, and P.Kumar, "Iterative Adaptive Switching Median Filter," ICICA-2006, pp.1024-1029, Jan. 2006.

[111] Haindi Ibrahim, Nichalas SiaPik Kong and Them Foo Ng, "Simple Adaptive Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," IEEE Trans. On Consumer Electronics,Vol. 54, No.4, pp. 1920-1927, Nov. 2008.

[112] Jianjun Zhang Q in Wang, "An Efficient Method for Removing Random –Valued Impulse noise," ICALP – 2008, pp. 918-922, July. 2008.

[113] Wenbin Luo and Dung Dong, "An Efficient Method for the Removal of Impulse Noise," ICIP-2006, pp.-2601-2604, May. 2006.

[114] C. L. Chan, A. K. Katsaggelos, and A. V. Sahakian, "Image Sequence Filtering in Quantum-Limited Noise with Applications to Low-Dose Fluoroscopy", IEEETransactions on Medical Imaging, Vol.12, No. 3, pp.610 – 621, September 1993.

[115] H. Soltanian-Zadeh, J.P. Windham, and A.E. Yagle, "A Multidimensional Nonlinear Edge-Preserving Filter for Magnetic Resonance Image Restoration*",* IEEE Transactions on Image Processing, vol. 4, no.2, pp.147 – 161, February 1995.

[116] J. A. Goyette, G. D. Lapin, M. G. Kang, and A. K. Katsaggelos, "Improving Autoradiograph Resolution Using Image Restoration Techniques", IEEEEngineering in Medicine Biology, vol. 10, pp. 571 – 574, August/September 1994.

[117] T.P. ORourke and R.L. Stevenson, "Improved Image Decompression for Reduced Transform Coding Artifacts", IEEE Transactions on Circuits and System for Video Technology, vol. 5, no.6, pp. 490 – 499, December 1995.

[118] S. Iyer and S. V. Gogawale, "Image Enhancement and Restoration Techniques in Digital Image Processing*",* Computer Society of India Communications Proceeding, pp. 6 – 14, June 1996.

[119] X. Xu and E. L. Miller, "Adaptive Two-Pass Median Filter to Remove Impulsive Noise",In Proceedings of International Conference on Image Processing 2002, pp. I–808 – I–811, September 2002.

[120] K. Kondo, M. Haseyama, and H. Kitajima, "An Accurate Noise Detector for Image Restoration", In Proceedings of International Conference on ImageProcessing 2002, vol.1, pp. I–321 – I–324, September 2002.

[121] C. Butakoff and I. Aizenberg, "Effective Impulse Detector Based on Rank- Order Criteria", IEEE Signal Processing Letters, vol. 11, no. 3, pp. 363 – 366, March 2004.

[122] V. Crnojevic, V. Senk, and Z. Trpovski, "Advanced Impulse Detection Based on Pixel-Wise MAD", IEEE Signal Processing Letters, vol. 11, no. 7,pp. 589 – 592, July 2004.

[123] E. Abreu and S. K. Mitra, "A Signal-Dependent Rank Ordered Mean (SDROM) filter-A New Approach For Removal Of Impulses From Highly Corrupted Images", In Proceedings of International Conference on Acoustics, Speech, andSignal Processing, vol. 4 of ICASSP-95, pp. 2371 – 2374, May 1995.

[124] L. Khriji and M. Gabbouj, "Median-Rational Hybrid Filters", In Proceedings ofInternational Conference on Image Processing 1998, vol. 2, pp. 853 –857, October 1998.

[125] Haidi Ibrahim, Nicholas Sia Pik Kong and Theam Foo Ng, "Simple Adaptive Median Filter for the Removal of Impulse Noise from Highly Corrupted Images", IEEE Trans. on Consumer Electronics, vol. 54, no. 4, November 2008.

[126] T. Ravi Kishore and K. Deergha Rao, "A Fast and Reliable Median Filter for extremely Corrupted Images",TENCON-2008,IEEE Conference, pp. 1-5, 2008 .

[127] Faouzi Alaya Cheikh, Ridha Hamila, Moncef Gabbouj and Jaakko Astola, "Implse Noise Removal In Highly Corrupted Images", IEEE Trans. Image Processing, 0-7803-3258-X/96,pp. 997-1000, 1996.

[128] S. K. Mitra, T. H. Yu, R. Ali, "Efficient Detail Preserving Method Of Impulsive Noise Removal From Highly Corrupted Images", SPIE Proc. On image and video Processing, pp. 43-48, 1994

[129] Vladimir Crnojevic, "Impulse Noise Filter With Adaptive MAD Based Threshold," IEEE International Conference on Image Processing, vol. 5, pp. 789-993, May – 2005.

[130] Tzu-choLin, "Progressive decision based Mean Type Filter for Image Noise Suppression," Computer Standard and Interfaces, vol. 30, pp. 106-114, Aug 2008.

[131] Pinor Civicioglu, Mustata Alci and Erkan Besdsok, "Impulse Noise Suppression from Image with the Noise Exclusive Filter," EURASIP Journal on Applied Signal Processing, vol. 16, pp. 2434-2440, 2004.

[132] Bogdam Smolka, "Adaptive Impulsive Noise in Color Images," proceeding of 2009 APSIPA Annual Summit and Conference, Sapporo, Japan, Oct4-7 2009.

[133] Z. Ma, D. Feng, H.R. Wu, "A Neighborhood Evaluated Adaptive Vector Filter for Suppression Of Impulse Noise In Color Images", Real-Time Imaging, vol. 11, no. 6, pp. 403-416, 2005.

[134] L. Khriji, M. Gabbouj, "Adaptive Fuzzy Order Statistics-Rational Hybrid Filters For Color Image Processing", Fuzzy Sets and System, vol. 128, no. 1, pp. 35-46, 2002.

[135] M. S. Moore, M. Gabbouj, and S. K. Mitra, "Vector SD-ROM Filter for Removal of Impulse Noise from Color Images," presented at the PECMCS EURASIP Conf. DSP for Multimedia Communications and Services, Poland, 1999.

[136] Z.Ma and H. R.Wu, "A Histogram Based Adaptive Vector Filter For Color Image Restoration," in Proc. 4th Pacific Rim Conf. Multimedia, vol. 1, pp. 81–85, 2003.

[137] Bogdon Smolka, Andrzej Chydzinski, "Fast Detection and Impulse Noise Removal in Color Images," Real-Time Imaging vol. 11, pp. 270-283, 2007.

[138] S.Md.Mansoor Roomi, T.Pandy Maheswari, V.AbhaiKumar, "A Detail Preserving Filter for Impulse Noise Detection and Removal," GVIP Journal, vol. 7, no. 6,pp. 8-3. Nov 2007.

[139] C.R.Vogel and M.E.Oman, "Fast, Robust Total Variation – Based Reconstruction of Noisy, blurred Images," IEEE Trans. on Image Processing, vol. 7, pp. 4-7, 1998.

[140] K.S.Srinivasan and D.Ebenezer, "A New Fast and Efficient Decision Based Algorithm for Removal of High – Density Impulse Noise," IEEE Signal Processing letters, vol. 14, no. 3, pp. 189-192, March 2007.

[141] X.D.Jing, "Image Detail Preserving Filter for Impulse Noise Attenuation," IEE Pro-vis. Image Signal Process. vol. 150, no. 3, June 2003.

[142] Yunhua Wang, Linda S. DeBrunner, Joseph P. Havlicek, and Dayong Zhou, "Signal Exclusive Adaptive Average Filter for Impulse Noise Suppression," IEEE Trans. on Image Processing, vol.6, no. 4, 2006.

[143] Raymond.H.Chan, Chen Hu, and Mila Nikalova, "An Iterative Procedure for Removing Random-Valued Impulse Noise," DRAFT, pp. 1 – 8, Feb-2004.

[144] Yiqiu Dong and Shufang Xu, "A New Directional Weighted Median Filter for Removal of Random-valued Impulse Noise," IEEE Signal Processing letters, vol. 14, no. 3, pp. 193-196, March 2007.

[145] Hancheng Yu, LiZhao and HaixianWang, "An Efficient Procedure for Removing Random-Valued Impulse Noise in Images," IEEE Signal Processing Letters, vol. 15, pp. 922-925, 2008.

[146] Gaihua Wang, DehuaLi,Weimin Pan, and ZhaoxiangZang, "Modified Switching Median Filter for Impulse Noise Removal," Signal Processing, vol. 90, no. 12, pp. 3213-3218, 2010.

[147] Ali Said Awad, Hong Man, "Similar Neighbor Criterion for Impulse Noise removal in Image," International Journal of Electronic Communication, vol. 64 , no. 10, pp. 904-915, 2010.

[148] G.Landi, "An Algorithm for Image Denoising With Automatic Noise Estimate," Journal of Mathematical Imaging and Vision, vol. 2, no. 1, pp. 98-106, 2009.

[149] X.M.Zhang, Z.P.Yin and Y.L.Xiong, "Adaptive Switching Mean Filter Using Conditional Morphological Noise Detection," Electronics letters, vol. 44, no. 6,pp. 986-987, March 2008.

[150] Tzu-ChaoLin, "Progressive Decision Based Mean Type Filter for Image Suppression," Computer Standard and Interference, vol. 30, pp. 106-114, 2008.

[151] Dung Dang, WenbinLao, "Impulse Noise Removal Utilizing Second-Order Difference Analysis," Signal Processing, vol. 87, pp. 2017-2025, 2007.

[152] Wenbin Luo, "Efficient Removal of Impulse Noise from Digital Images," IEEE Trans. on consumer electronics, pp. 523-527, May 2006.

[153] A.S Awad and H.Man, "High Performance Detection Filter for Impulse Noise Removal in Images," Electronics Letters, vol. 44, no. 3, Jan 2008.

[154] Chung-Chaia Kang, Wen-JuneWang, "Modified Switching Median Filter with one More Noise detection for Impulse Noise Removal," International Journal of Electronics and Communication, 2008.

[155] Rabic, "Robust Estimation Approach for Blind Denoising," IEEE Trans. on Image Processing, vol. 14, no. 11, pp. 1755-1765, 2005.

[156] M.Black and A.Rangreajan, 'On the Unification of Line Process Outlier Rejection and Robust Statistics with Applications to Early Vision," International Journal of Computer vision, vol. 19, pp. 57-91, 1996.

[157] S.Sardy,P.Tseng and A.Broce, "Robust Wavelet Denoising," IEEE Signal Processing, vol. 49, no. 6, pp. 1146-1152, Jan. 2001.

[158] M.B Meenovathi and K.Rajesh, "Vulture Filtering Techniques for Removal of Gaussian and Mixed Gaussian – Impulse Noise," World Academy of Science, Engineering and Technology, vol. 26, pp. 238-244, 2007.

[159] Haidi Ibrahim, Nicholas Sia Pik Kong and Theam Foo Ng, "Simple Adaptive Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," IEEE Trans. on consumer electronics, vol. 54, no. 4, pp. 1920-19277, 2007.

[160] S.Indu, Chaveli Ramesh, "A Noise Fading Technique for Images Highly Corrupted with Impulse Noise," International Conference on Computing Theory and Application (ICCTA-2007), pp. 627-632, 2007.

[161] Gaohang Yu, Jinhong Huong and Yizhou, "A Descent Spectral Conjugate gradient Method for Impulse noise Removal," Applied Mathematic Letters, vol. 23, no. 5, pp. 555-560, May 2010.

[162] T.K.Thivakaran, R.M. Chandrasekaran, "Non linear Filter Based Image Denoising usingAMF Approach," IJCSIS, vol. 7,no. 2,pp. 224-227, 2010.

[163] Chin-Chen Chang, Ju-Yuan Hsiao Chinh-Ping Hsieh, "An Adaptive Median Filter for Impulse denoising," IITA, vol. 2, pp. 346-350, 2008.

[164] Rang-HongHuang and xiao-hongWang, "Image Denoising Via Gradient Approximation by Up wind Scheme," Signal Processing (Elsevire), vol. 88, no. 1, pp. 69-74, Jan. 2008.

[165] J.Harikiran, B.Saichandana, and B.Divakar, "Impulse Noise Removal in Digital Images," International Journal of Computer Application, vol. 10, no. 1, pp. 39-42, Nov. 2010.

[166] Y.Zhang, Y Chaisah, W Chen, "Robust Information Filter for Decentralized Estimation," vol. 41,no. 1, pp. 2141-2146,Dec. 2005.

[167] WeiHuang, Jiang Bo, "Adaptive Threshold median Filter for Multiple-Impulse Noise", Journal of Electronic Science and Technology of China. Vol.5 no.1, pp.1-5, 2007

[168] P.G.J Barten, "Contrast Sensitivity of Human Eye and Its Effects on Image Quality", SPIE, Washington, 1999.

[169] M. Marta, S. Grgic, M. Grgic, "Picture Quality Measures In Image Compression Systems", Proceedings EUROCON '03, p. 233-7, 2003.

[170] Z. Wang, A.C. Bovik, "A Universal Image Quality Index", IEEE Signal Processing Letters, vol. 9, no. 3, pp.81-84, 2002.

[171] K.S.Srinivasan, D.Ebenezer, "A New Fast and Efficient Decision-based Algorithm for Removal of High-Density Impulse Noise", IEEE Signal Processing Letters, vol. 14, no. 3, pp.189-192, march-2007.

[172] Chen Cong-ping ; Wang Jian ; Qin Wu ; Dong Xiao-gang, "A new adaptive weight algorithm for salt and pepper noise removal", Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference, pp.26-29,April-2011.

[173] Mélange, T.; Nachtegael,M.;Kerre, E.E.; Fuzzy Random Impulse Noise Removal From Color Image Sequences , Image Processing, IEEE Transactions, Issue:4, pp.959 – 970, April 2011

[174] Tom Mélange, Mike Nachtegael and Etienne E. Kerre, "Random impulse noise removal from image sequences based on fuzzy logic", J. Electron. Imaging 20, 013024 (Mar 25, 2011); doi:10.1117/1.3564922

Dissemination of Research Outcome

(Explicitly Mentioned in Thesis)

[P1] **Ramesh Kulkarni**, Manoj Ku. Gupta,Sukadev Meher, "An Efficient Detail Preserving Adaptive Noise Detection and Suppression Filter for Impulse Noise", *ICCCI-09* at VESIT, Mumbai, Jan2-3, 2009.

[P2] **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "An Algorithm for Image Denoising by Robust Estimator," *EJSR*, vol. 39, no. 3, pp. 372-380, 2010.

[P3] **R.K.Kulkarni**, S.Meher,C.B.Lohoti, "Impulse Denoising Using Improved Progressive Switching Median Filter", *International Conference & Workshop on Emerging trends in Technology 2010* , Mumbai, Proceeding is published by ACM publication & papers are put online in Digital Lab.. Feb 26 – 27, 2010.

[P4] **R.K.Kulkarni**, S.Meher, J.M.Nair, "Impulse Noise Removal in Highly Corrupted Image by Impulse Classification", International Conference on Computational Intelligence *Applications2010, ICCIA*- 2010, held on 03-05 March 2010 in SITRC, Nasik, sponsored by University of Pune.

[P5] **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "Removal of Impulse Noise from Highly Corrupted Images By Adaptive Switching Filter", International Conference on Electronic Systems, NIT Rourkela-769 008, Orissa, India, Jan-2011.

[P6] **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "An Adaptive Switching Filter for Removing Impulse Noise from Highly corrupted Images," *GVIP*, ACM Publication,vol. 10,no. 4,pp. 47-53,oct-2010.

[P7] **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "Impulse Denoising Using Adaptive Switching Filter", Communicated to International Journal Signal Processing, Springer publication.

[P8] **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "A New Algorithm for Removal of Fixed and Random-Valued Impulse Noise From Corrupted Images," *International Conference on Emerging Trends in Computer Science, Communication & Information Tech.* held on Jan 9-11,2010 in Nanded, Maharashtra, India.

[P9] **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "Adaptive Local Thresholding with MAD (ALT-MAD) Algorithm", Communicated to International Journal Signal Image Video Processing, Springer publication.

(Not Explicitly Mentioned in Thesis)

International Journals:

1. **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "An Algorithm For Impulse Noise Reduction By Robust Estimator In Highly Corrupted Images," *Journal of Theoretical and Applied Information Technology*, vol. 10,no. 1, pp.9-15,Dec. 2009.
2. **R.K.Kulkarni**, S.Meher**,** J.M.Nair "An Effective Approach for suppressing High Density Noise in Image by Robust Estimator," *Journal of Advance in Engineering Science*, Section-*c*, vol.- 8,pp.39-46,2010

International Conference

1. **R.K.Kulkarni,** S.Meher,Vidya.S.Lunge, "Dumpster-Shafer theory for image restoration", *International Conference & Workshop on Emerging trends in Technology 2010* , Mumbai, Proceeding is published by ACM publication & papers are put online in Digital Lab.. Feb 26 – 27, 2010.
2. **R. K. Kulkarni**, S. Meher, J.M.Nair,"Efficient Removal of Impulse Noise Using Adaptive Median Filter", 'NUiCONE – 2010', International Conference On Current Trends In Technology, Institute of Technology, Nirma University, Ahmedabad – 382481, 9-11 December, 2010.

National Conferences:

1. **R.K.Kulkarni**,S.Meher,J.M.Nair, "Adaptive Noise Detection and Suppression Filter for Impulse Noise", *2nd National conference Recent Trends in Computer Engineering (RTCE-09)* on 30th Dec-2009, at Sinhgad College of Engieering,Pune-41
2. **R.K Kulkarni,** S.Meher,C.B.Lohoti ,"Comparative study of Median based Nonlinear Filters for Impulse Denoising", *National Conference on Recent Trends in Engineering & Technology*, held on Oct-30-31st-2009 in Vashi, Navi – Mumbai.
3. **R.K.Kulkarni**, S.Meher, J.M.Nair, "Adaptive Median Filter for Impulse Noise from Highly Corrupted Images", *National Conference on New Advances in Core Computing and their challenges*. March-20 – 21, 2010, at M.B.M Engineering Collage, JNV University Jodhpur, sponsored by UGC New Delhi.
4. **R.K.Kulkarni**, S.Meher**,** J.M.Nair, "Removal of Impulse Noise in Color Images", *equinox-2011*, National conference, Sep-2011.